PSRT

IITiS

Proceedings of

# 14th

# Polish Teletraffic Symposium

Zakopane, 20-21 September 2007

Proceedings of

# 14th Polish Teletraffic Symposium

**Zakopane, 20-21 September 2007**


**EDITOR**

**Tadeusz Czachórski**

**Gliwice 2007**

## PUBLISHED BY

**Institute of Theoretical and Applied Informatics**

**of the Polish Academy of Sciences**

**Bałtycka 5, 44-100 Gliwice, POLAND**

**www.iitis.gliwice.pl**

## TECHNICAL PROGRAM COMMITTEE



## ORGANIZING COMMITTEE

Chair: **Krzysztof Grochla**
pts@iitis.gliwice.pl,
Phone: +48 32 231 73 19 ext 218; Fax: +48 32 231 70 26

**Joanna Domańska**
**Sławomir Nowak**

Cover designed by Krzysztof Grochla

## ISBN: 978-83-926054-0-9

# CONTENTS

## KEYNOTE TALK



## QOS



## SWITCHING



## CELLULAR AND WIRELESS NETWORKS

## TRAFFIC ENGINEERING AND ROUTING



## TOOLS AND MODELS

# PREFACE

The proceedings of the 14th Polish Teletraffic Symposium, held in Zakopane, 20-21 September 2007 contain 28 contributions refereed and selected by its Technical Programme Committee.

The symposium has been organised since 1994 on a rotational basis by the following Polish universities and institutes:

- Poznań University of Technology, Poznań;
- Gdańsk University of Technology, Gdańsk;
- University of Mining and Metallurgy, Cracow;
- Warsaw University of Technology, Warsaw;
- Wrocław University of Technology, Wrocław;
- Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice.

PTS 2007 was organized by IITiS PAN. Recently, each second year, PTS becomes Polish-German Teletraffic Symposium with participation of Dresden University of Technology, Dresden

The aim of the Symposium is to provide a regular, annual forum for a discussion on the design, implementation and perfection of contemporary telecommunications and computer communications systems. Its areas of interest include

- Teletraffic Theory, Modelling and Optimisation
- Performance Evaluation and Modelling of Communication Protocols
- Traffic Measurements
- Traffic Issues for Mobile Systems
- Mobility Models
- Traffic Issues for Internet
- Traffic Issues for High Speed, Packet-Switched Networks
- Traffic Models for Optical Networks
- Performance Analysis Methods and Simulation
- Intelligent Routing Protocols
- Broadcast and Multicast Traffic Control and Management
- QoS Issues

The Symposium is addressed mainly to the teletraffic and network design community, both in Academia and Industry.

For the Technical program committee of PTS 2007

**TADEUSZ CZACHÓRSKI**
EDITOR OF THE PROCEEDINGS

Polish Teletraffic Symposium
2007
ISBN 978-83-926054-0-9

# KEYNOTE TALK

## On providing QoS in the Internet

WOJCIECH BURAKOWSKI [a] HALINA TARASIUK [b]

Institute of Telecommunications
Warsaw University of Technology
[a] *wojtek@tele.pw.edu.pl*
[b] *halina@tele.pw.edu.pl*

The Internet in today does not satisfy users, service providers and infrastructure operators. In Europe, except research networks as GEANT or some NRENs (National and Research National Networks), the *best effort* service is still only one network service offered to the users. The lack of offering QoS (*Quality of Service*) capabilities of the network becomes one of critical barriers in using more rigorous applications than e.g. web services. One can also mention about other barriers in the Internet as e.g. security, supporting mobility or reliability. The drawbacks of the Internet are well recognized but, at least for now, there are not satisfactory solutions for solving the above mentioned problems. Nowadays, there is discussion around Europe (around the whole world) about Future Internet and one can find interesting points expressed in documents of EIFFEL or FIRE initiatives. Furthermore, inside FP7 Framework and ICT (*Information and Communication Technologies*) area the problem of Future Internet is emphasized.

This paper presents tested approaches as proposed for assuring QoS in the Internet, including such issues as end-to-end QoS, QoS architectures, multi-domain aspects, signaling etc. In particular, we focus on the solutions tested inside FP5 and FP6 projects as AQUILA and EuQoS. While AQUILA solution follows the DiffServ Architecture proposed for single domain, the EuQoS approach can be regarded as implementation of the IMS/TISPAN architecture proposed for Next Generation Networks including multi-domain aspects.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 9 - 20

# Measurements for on-line QoS monitoring of Real Time Class of Service in IP networks

PIOTR STASIEWICZ [a] WOJCIECH BURAKOWSKI [b]

Institute of Telecommunications
Warsaw University of Technology
[a] *piotr.stasiewicz@elka.pw.edu.pl*
[b] *wojtek@tele.pw.edu.pl*

**Abstract:** The paper shows how to perform on-line QoS (Quality of Service) monitoring of Real Time Class of Service (RT CoS) in IP QoS networks. The RT CoS is aimed at handling streaming traffic that is generated by applications as VoIP, VTC etc. For this class of service we assume that the volume of submitted traffic is limited by appropriate CAC (Connection Admission Control). In the described method we monitor the QoS under maintaining the maximum allowed the worst-case traffic conditions as controlled by CAC. Since such conditions occur only sometimes (only when the maximum allowed volume of payload traffic is submitted), in our approach we keep these conditions by introducing additional traffic to the network. The volume of this traffic is the difference between the maximum allowed traffic and the payload one. The paper describes the details of the method including implementation issues and numerical results (simulation and experimental) showing its performances.
**Keywords:** QoS, measurements, monitoring, Real Time CoS

## 1. Introduction

A pressure for introducing QoS (Quality of Service) into the Internet is coming from the service providers who want to get a profit from new real-time services as VoIP, VTC, tele-medicine, etc. However, to meet these expectations many technical problems still need to be solved, mainly related to network traffic control and Internet signaling. Anyway, one can expect that the market solution for QoS Internet will appear soon. Nowadays, a number of EU IST projects announce promising solutions that are currently tested in some prototype networks, as GEANT [3], EuQoS [1], DAIDALOS [2], MUSE [4], OPERA [5] etc. In this paper we focus on the problem of on-line QoS monitoring offered by the network that is important for both the operators and the users. The operators of QoS network require knowledge about the state of the offered by them QoS level while the users expect the confirmation from the network operator about the received QoS. We assume that for providing QoS into the Internet, we implement the Class of Services concept, as in EuQoS system described e.g. in [6].

In this paper we present a method for on-line monitoring of QoS level offered by a Class of Service (CoS) in IP-based network that supports real-time applications like VoIP, VTC, named Real Time CoS (RT CoS). The method is aimed for providing knowledge to the network operator about the QoS offered by discussed CoS. More particularly, it should allow us for measuring the values of parameters as IPTD (IP Packet Transfer Delay), IPLR (IP Packet Loss Ratio) and IPDV (IP Delay Variation) that characterize the QoS level [8].

Furthermore, the values of the mentioned parameters should be measured under, so called, the "worst-case" traffic conditions and it means that traffic conditions should be maintained as adequate for the case, when amount of admitted traffic gets the upper limit. To keep during the measurements the worst case traffic conditions is very important since the QoS offered by the CoS, characterized in terms of IPTD, IPDV and IPLR values, is specified under assuming these traffic conditions. For instance, if we measure QoS when only a few connections is running we may expect better quality than assumed by the CoS but these measurements say nothing about the QoS offered by this CoS. The rest of the paper is structured as follows. In section 2 we present the method for on-line monitoring. Then, in section 3 we briefly describe the main objectives for RT CoS that is discussed as an example for on-line monitoring. Next, in section 4 we include exemplary numerical results showing effectiveness of the approach. Finally, section 5 summarizes the paper.

## 2. Method for on-line monitoring of QoS offered by Class of Service

Let us recall that Class of Service is the term used for the first time by ATM people for defining specific treatment of ATM cells generated by given types of applications. For ATM, we called them as ATM Native Services, and in this spirit have been defined the services as CBR, rt-VBR, nrt-VBR, ABR, UBR and GFR. In the ATM switches, for each of the mentioned services we dedicate the separate queues with specific packet scheduling disciplines as PQ (Priority Queue) or WFQ (Weighted Fair Queuing). The forced approach for the Internet with QoS is to follow the same direction. As a consequence, the IETF has proposed a number of Classes of Services [7]. Among them, the RT CoS is specified.

For RT CoS we dedicate in each link an amount of bandwidth, say C. The issue is to limit the volume of traffic submitted to this CoS by using appropriate Admission Control (AC) rules. In this way, we can control the values of parameters characterizing the packet transfer that is described by the values of the parameters as IPTD, IPDV and IPLR. So, the monitoring of the CoS is to measure, possibly on-line, the values of the mentioned parameters. For this purpose, we need to submit to this CoS additional traffic, named measurement traffic (MT), just for providing the measurements. On the other hand, we need to maintain the volume of traffic handled by the CoS on the same level independently of the volume of the payload traffic. This requirement comes from a need for keeping the CoS on the maximum load allowed by the AC rules. Anyway, it needs to introduce again to the network additional traffic, named background traffic (BT), now related with the payload traffic (PT). Of course, we should keep that BT+PT = *constant*.

The scheme for providing on-line monitoring in the case of inter-domain link is depicted on Fig.1. We assume that the Borders Routers (BR) provide QoS mechanisms at the packet level, i.e. PHB (*Per Hop Behavior*) mechanisms as classifiers, schedulers etc. Consequently, in general case, a number of CoSs can be implemented, among them the discussed RT CoS. For RT-CoS we dedicate separate buffer and we believe that this class is handled in isolation of the other CoSs, if any. Therefore, we put our attention on this CoS only and we do not consider the presence of other traffic. The plan for deploying the traffic generators and traffic receivers in the case of monitoring RT CoS on inter-domain link is also shown in Fig.1. The MT traffic is generated by the MTG (*Measurement Traffic Generator*) and is received by the MTR (*Measurement Traffic Receiver*). The MTG traffic is submitted to an input port of BR and this traffic is received by the MTR connected to the port of the BR terminating RT CoS. Similarly, the BT traffic is handled. Remark, that for

MT and BT we will use the same ports in BRs. On the other hand, we do nothing with PT traffic.

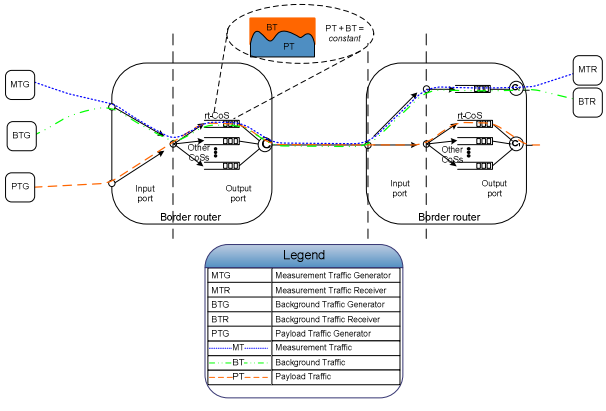



Fig.1 The measurement scheme for "on line monitoring"

## 3. Real Time Class of Service

In this section we briefly outline the main characteristics for RT CoS. As we mentioned above, the RT CoS is designed for transferring the streaming traffic emitted by such applications as VoIP and VTC. More details about the AC rules for the RT CoS class one can find e.g. in [9]. The traffic submitted to the RT CoS is handled independently on the other traffic and, for this purpose in the routers we dedicate for it a separate queue. The size of the queue is short since we need to keep low values of IPTD (say 100 ms) and IPDV (say 50 ms). For maintaining low value of IPLR (say $10^{-3}$) we assume special AC algorithm for limiting volume of submitted traffic. Furthermore, the RT CoS traffic is handled with high priority. The AC algorithm for this CoS assumes that particular connections produce CBR (Constant Bit Rate) traffic that is policed at the network entry point.

| ***Time [s]*** | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Number of calls*** | 60 | 70 | 80 | 90 | 105 | 112 | 120 | 120 | 125 | 90 | 75 | 60 |
| ***Payload [kbit/s]*** | 3840 | 4480 | 5120 | 5760 | 6720 | 7168 | 7680 | 7680 | 8000 | 5760 | 4800 | 3840 |
| ***Measurement traffic [kbit/s]*** | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| ***Background traffic [kbit/s]*** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1575 | 1255 | 3495 | 4455 | 5415 |
| ***IPLR*** | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0078 | 0,0075 | 0,0078 | 0,0083 | 0,0083 |

Fig.2 Schedule of the simulation in scenario #1; monitored link capacity: 10Mbit/s, length of the buffer in access node: 10packets

## 4. Numerical results

In this section we present results of two simulation experiments that show effectiveness of our on-line monitoring method of RT CoS. The objective of the tests was to validate assumptions of the measurement method. The architecture of simulated system was described in detail in section 2. In first scenario method was used to detect

degradation of QoS level what was caused by wrong configuration of admission control mechanism. Second scenario shows how the method can be used to detect problem caused by users who abuse declared traffic contract and generate greater volume of traffic than they supposed to.

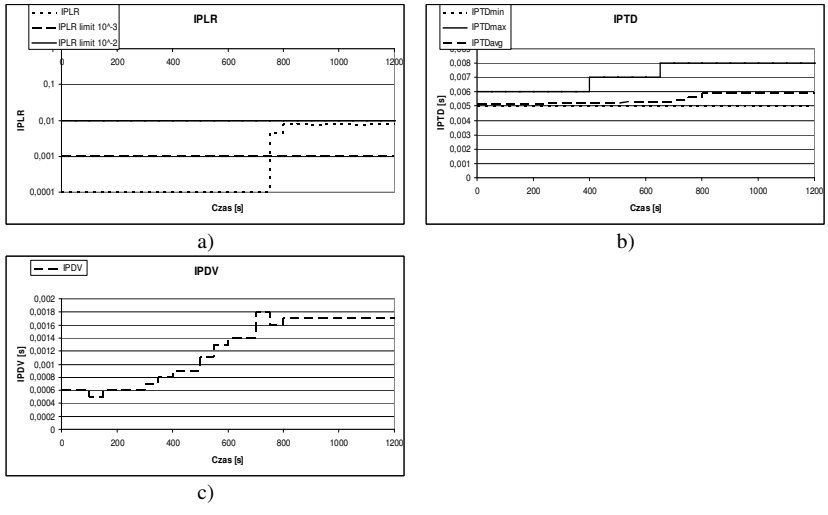



Fig.3 Scenario #1 – results; monitored link capacity: 10Mbit/s, length of the buffer in access node: 10packets; a) IPLR, b) IPTD, c) IPDV

### 4.1 Scenario #1

In this scenario we were simulating wrong configuration of the admission control algorithm for the Real Time CoS in access router. This class of service, as we mentioned before, requires very low values of IPLR (less then $10^{-3}$). To keep low values of IPLR and IPDV buffer length was set to 5 and 10 packet respectively. Configuration was corrupted and AC limit was set to ensure loss ratio less then $10^{-2}$ (10 times higher than assumed).

Size of the payload and background traffic packets was 200 bytes. Single source generates packets at equal intervals, every 25ms. In simulation CBR sources were replaced with one Poisson source that was sending packets with corresponding bit rate. Measurement packets were sent every 4ms and were 100 bytes long (total bit rate 200kbit/s).

During the simulation number of active calls was increasing. Till the background traffic was off and payload was low, QoS level was granted. However since background traffic generator has started and resources were filled up to wrong AC limit, QoS degradation was observed and IPLR rapidly increased. Pre-defined schedule of the simulation is shown on Fig.2 and results are presented on Fig.3.

### 4.2 Scenario #2

In this scenario we were using on-line monitoring method to detect degradation of QoS in case when end-users abuse declared traffic contracts. Such behavior in situation when amount of admitted traffic gets the upper limit and link utilization approaches allowed maximum may cause degradation of QoS level from the point of view of other

users. Simulation model was equal to this used in previous paragraph. Due to low delay restrictions, buffer length was set to 5 or 10 packets respectively.

In first part of scenario number of running calls was lower then maximal admissible. However total payload traffic was higher then declared one and since background traffic appeared in network, threshold for IPLR was overflowed. Number of active calls was constant during the test and summary declared load was equal to ¾ of available resources. In first phase generated payload traffic was higher then declared by 20%, in second by 10% and in last one – third by 5%. Fig.4 and Fig.5 present schedule of scenario #2 and measured results. One must notice that real reason of the QoS level degradation in this case can not be detect by our method. Equal symptoms could occur in situation when CAC mechanism is wrong configured. However crossing the threshold value in the worst-case traffic conditions should be sufficient signal for administrator to check configuration of CAC or policing mechanism.

| ***Phase*** | **I** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ***Time [s]*** | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 |
| ***Number of calls*** | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ***Declared payload [kb/s]*** | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 |
| ***Generated payload [kb/s]*** | 7255 | 7255 | 7255 | 7255 | 7255 | 7255 | 7255 | 7255 |
| ***Measurement traffic[kb/s]*** | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| ***Background traffic[kbis]*** | 0 | 0 | 0 | 0 | 2150 | 2150 | 2150 | 2150 |
| ***IPLR*** | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0118 | 0,0121 | 0,0127 |
| | | | | | | | | |
| ***Phase*** | **II** | | | | **III** | | | |
| ***Time [s]*** | 450 | 500 | 550 | 600 | 650 | 700 | 750 | 800 |
| ***Number of calls*** | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ***Declared payload [kb/s]*** | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 |
| ***Generated payload [kb/s]*** | 7040 | 7040 | 7040 | 7040 | 6720 | 6720 | 6720 | 6720 |
| ***Measurement traffic [kb/s]*** | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| ***Background traffic[kb/s]*** | 2150 | 2150 | 2150 | 2150 | 2150 | 2150 | 2150 | 2150 |
| ***IPLR*** | 0,0093 | 0,0052 | 0,0048 | 0,0047 | 0,0032 | 0,0019 | 0,0023 | 0,0024 |

Fig.4 Scenario #2, monitored link capacity: 10Mbit/s, length of the buffer in access node: 10packets

### 4.3 Conclusions

Presented results show clearly that our method could be used for detecting errors in configuration of elements in network infrastructure responsible for providing QoS. It could be performed both as on-line monitoring process when probing packets are sent parallel to payload traffic and as testing process run after for example router reconfiguration when all volume of traffic is generated artificially by Traffic Generators to check new settings.

Second scenario proved also that our method is also useful for detecting QoS degradation in situation when users abuse their traffic contracts. Even small amount of unprovisioned traffic in case when link utilization gets the allowed maximum could cause QoS threshold level overflow. However wrong configuration of CAC or Policer can produce similar symptoms and can not be identified in such way but it is sufficient signal to find the reason of the QoS violation.

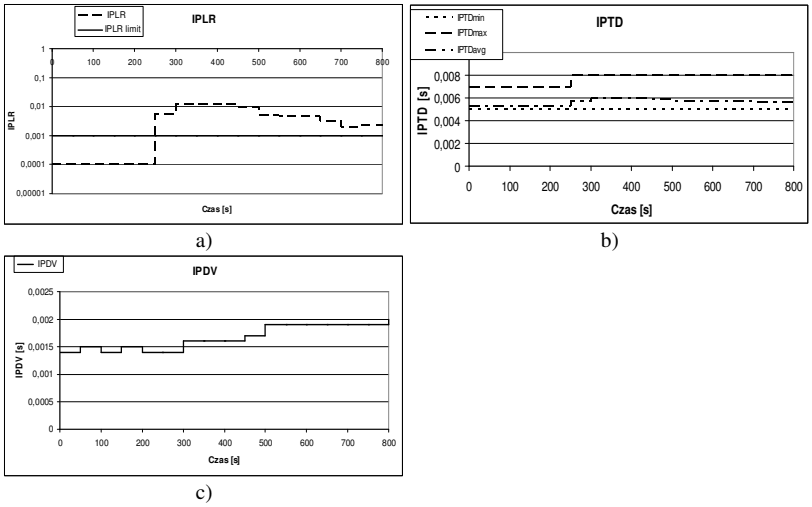



Fig.5 Scenario #2 – results; monitored link speed: 10Mbit/s, length of the buffer in access node: 10packets; a) IPLR, b) IPTD, c) IPDV

## 5. Implementation details

In this section we briefly describe implementation details of on-line monitoring method. According to assumptions implementation was integrated with EuQoS framework. It consists of three separate components: Management Console (MC), Traffic Generating Controller (TGC) – module that is integral part in Resource Allocator (RA), developed in EuQoS Project, and a couple Traffic Generator (TG) and Traffic Receiver (TR). Communication between modules is based on remote procedure call standard, and based on this mechanism XML-RPC protocol [10][11], that is safe and easy way to exchange small amount of data between distributed applications. Architecture is presented on Fig.6.

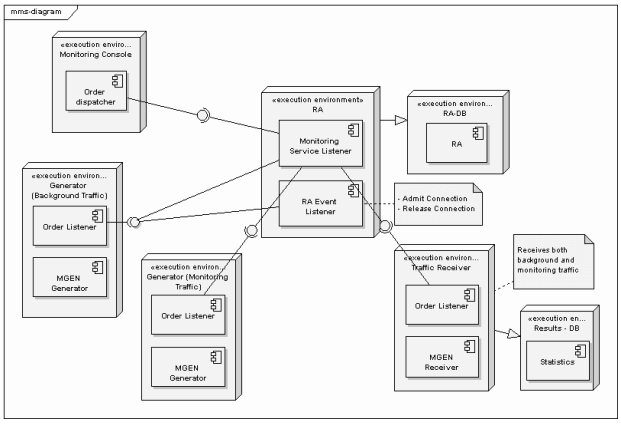



Fig.6 Architecture of on-line monitoring method components

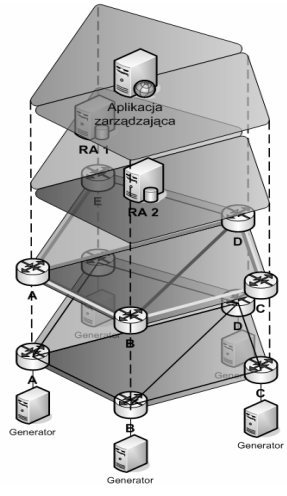
Assumptions that were considered during analysis and design process were as follow:

- fully integration with EuQoS system,
- every module must enable save interfaces for communication,
- communication is realized via XML-RPC protocol,
- modules should run on Windows or Linux platforms,
- measurements must not cause QoS level degradation from point of view of other users,
- test results should be presented on-line, and stored in database,
- MGEN tool may be used as measurement and background traffic generator.

Below we briefly describe each component and monitoring process.

### 5.1 Management Console – MC

It is main component that controls measurements, collects and stores results. This Java application works also as graphical user interface. Configuration and results are stored in PostgreSQL data base [12]. Communication between MC and other components is based on XML-RPC protocol.




Configuration of the network and monitoring process is divided into four logical levels as on Fig.below. The highest level is MC that communicates with TGCs located in RAs (level 2). As we know one RA can be related with many physical links. Links can be grouped by users into so called relations that correspond to logical paths between network nodes. Those relations are placed in our order on third position under RAs. Relation can consist of links that belong to different domains, but also can contain only one link. Physical links are located directly under relations and are related with TGs and TRs which are connected to the source and destination node of each monitored link. One TG can generate traffic in many links and CoSs and one TR can receive packets from many links and CoSs.

MC can be used in both centralized and decentralized measurement management model. In centralized architecture main component is responsible for measurements in entire network and integrates communication between all probing agents. As an example we can mention measurement system developed by *Reseaux IP Eurepean Network Coordination Center – RIPE NCC* [13][14][15]. In decentralized model management components are distributed in autonomous domains that together build the network.

### 5.2 Traffic Generation Controller – TGC

The component is an integral entity of the Resource Allocator. It operates as a command dispatcher between Management Console, Traffic Generators and Receivers. It also controls process of generating background traffic in every CoS supported on given link. Because of the second function it requires actual information about available resources and needs to be triggered during processes that are related with resource release, allocation or reallocation. Every such event must be immediately propagated to adequate TGs to decrease or increase generated load. Additional traffic must not reduce

transmission quality observed by end users and traffic condition should be maintained as in the worst case traffic scenario for sufficient long period to keep measurement accuracy.

Original objects in RA that are responsible for CAC algorithms were extended with new function that returns a list object of all available for given link CoSs and available resources in each one from the point of view of CAC algorithm. RA sends returned by procedure values to the corresponding TGs where generated traffic volume must be changed.

### 5.3 Traffic Generator – TG

TG is an application developed in Python programming language. It generates background and monitoring traffic according to commands received from TGC. TG is fully controlled by TGC installed in RA, that sends information about monitored link, CoS and bit rate of the additional traffic.

Application implements interfaces to communicate with Multi-Generator MGEN, which is used to generate background traffic. Python enables comforTab.and easy way to run operating system commands that are used to start/stop MGEN and to system processes list, that is useful for proper management of MGEN instances. TG reads configuration from files that contains information about links and supported on them CoSs. The way of generating background traffic in given links and CoS is also specified in configuration. TG starts XML-RPC server that listens on commands sent by TGC. On the other hand TG periodically invokes “keep-alive” procedure on TGC to monitor signalization link. In case of error or exception during invoking that procedure, all traffic generators are to be disabled to avoid probability of link overload. While the program is running configuration of links, CoSs and MGEN instances is kept in the tree data structure, in which links are local roots and CoSs with assigned to them MGEN instances are leafs.

### 5.4 Traffic Receiver – TR

TR is an application developed in Java programming language. It receives both measurement and background traffic. While background packets are simply discarded, probing packets are analyzed. QoS parameters are being computed on the basis of those calculations.

Since TR must perform a lot of incoming data, its logic is based on ring buffer algorithm. All received probes are stored in list structures that are changed every equal period of time according to configuration and measurement window. This time period is also defined as results refresh time. Separate thread uses probes stored in lists to calculate QoS parameters. After this statistics are sent to Management Console via TGC, list is being cleaned and prepared for use in next loop.

### 5.5 Monitoring process

Monitoring process starts after administrator selects relations or single links in GUI and presses start button. Relation is then divided into single links and these are grouped by superior RAs. Next MC sends to each selected RA command to start monitoring of listed links. TGC module that is located in RA receives order, gets from CAC module information about available resources in given links and forwards command to adequate TGs and TRs. TG checks if given link is not already monitored. If so it only stores this information, otherwise it prepares configuration script for MGEN and runs new instance.

In TR configuration is extended with information about new source of monitoring and background packets. Then process that computes QoS parameters sends back results to TGC and it forwards them to MC where they are presented and stored in data base.

## 6. Experimental results

In this section we present the results of several practical experiments performed with our on-line monitoring method. The goal of the tests was to validate the design and implementation of our system. Two of them were repetitions of scenarios considered during the simulations, described in section 4. Testbed used in experiments was part of EuQoS experimental network. During the tests results and configuration commands were sent via other links than monitored ones and did not influence on monitoring process.

### 6.1 Scenario #1

This scenario was considered to show that on-line monitoring method does not cause degradation of QoS level from the point of view of the users in case when all network elements and mechanisms are properly configured and other users do not abuse their traffic contracts. During the test number of active calls was changing. Every event of connection release or admission triggered reconfiguration of Traffic Generator. Fig.7-a depicts amount of generated payload traffic and background traffic that sum to constant value of AC limit. Whereas Fig.7-b shows measured level of IPLR that despite of varying volume of payload and background traffic is under threshold all the time.

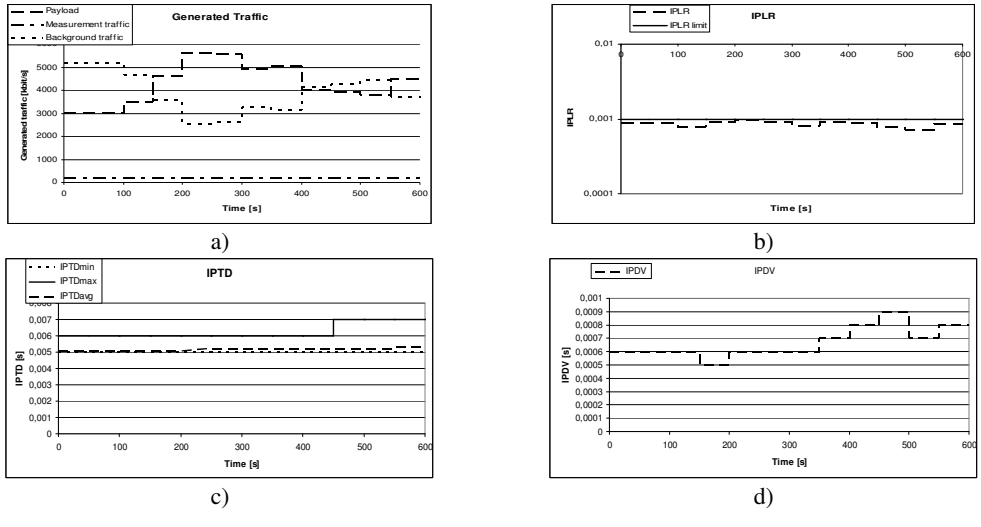



Fig.7 Scenario #1 – results; monitored link capacity: 10Mbit/s, length of the buffer in access node: 10packets; a) Generated traffic, b) IPLR, c) IPTD, d)IPDV

That results show, that on-line monitoring does not have any negative influence on QoS level noticed from the point of view of the users. Fig.7-a shows that summary volume of payload and background traffic is constant apart from the call-level events,

such as call admission or release and measurement period is long enough for accurate calculations.

### 6.2 Scenario #2

Next test was the second approach to the scenario already tested during simulations. In this scenario configuration of the CAC mechanism was corrupted and number of admitted calls was higher then assumed limit.

| *Time [s]* | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Number of calls* | 60 | 70 | 80 | 90 | 105 | 112 | 120 | 120 | 125 | 90 | 75 | 60 |
| *Payload [kb/s]* | 3840 | 4480 | 5120 | 5760 | 6720 | 7168 | 7680 | 7680 | 8000 | 5760 | 4800 | 3840 |
| *Measurement traffic [kb/s]* | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| *Background traffic [kb/s]* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1575 | 1255 | 3495 | 4455 | 5415 |
| *IPLR* | 0,0001 | 0,0001 | 0,0001 | 0,0001 | 0,0002 | 0,0005 | 0,0004 | 0,0020 | 0,0078 | 0,0089 | 0,0083 | 0,0092 |

Fig.8 Scenario #2; monitored link speed: 10Mbit/s; length of the buffer in access node: 10packets

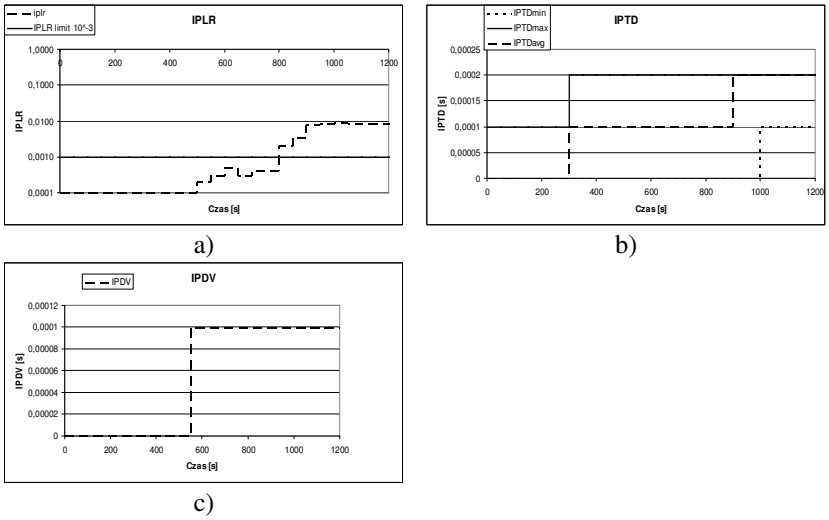



Fig.9 Scenario #2 – results; monitored link speed: 10Mbit/s; length of the buffer in access node: 10packets; a) IPLR, b) IPTD, c) IPDV

Presented results show, that wrong configuration of CAC mechanism would not be noticed as long as volume of traffic generated by users was below threshold of guaranty of appropriate QoS level. In reality notification about QoS degradation would be send in situation when many users were connected. Our method for on-line monitoring detected QoS violation after one measurement period in situation when number of active flows was low.

### 6.3 Scenario #3

Following scenario was also already analyzed in section 4 during the simulations. This test was considered to present usability of method for on-line monitoring of Real Time CoS for detection of QoS level degradation in situation when users abuse their traffic

contracts. That is possible when policing mechanism does not work properly and users generate greater load than they formerly declared. Bigger amount of traffic than declared in case when network utilization gets the allowed maximum can cause capacity and efficiency problems.

Test scenario was divided into three parts. There were only payload (greater than declared) and monitoring traffic in first phase. Volume of the extra traffic was about 4% of the available resources. In the second phase background traffic generator was activated. Generated load was the difference between available resources (got from CAC algorithm) and declared payload. During third part volume of extra payload decreased to about 2% of the available resources.

| ***Phase*** | **I** | | | | **II** | | | | **III** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Time [s]*** | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 | 550 | 600 |
| ***Number of calls*** | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ***Declared payload [kb/s]*** | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 | 6400 |
| ***Generated payload [kb/s]*** | 6720 | 6720 | 6720 | 6720 | 6720 | 6720 | 6720 | 6720 | 6560 | 6560 | 6560 | 6560 |
| ***Measurement traffic [kb/s]*** | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 |
| ***Background traffic [kb/s]*** | 0 | 0 | 0 | 0 | 1960 | 1960 | 1960 | 1960 | 1960 | 1960 | 1960 | 1960 |
| ***IPLR*** | 0,0001 | 0,0003 | 0,0002 | 0,0001 | 0,0016 | 0,0022 | 0,0027 | 0,0029 | 0,0022 | 0,0019 | 0,0017 | 0,0018 |

Fig.10 Scenario #3, monitored link speed: 10Mbit/s; length of the buffer in access node: 10packets

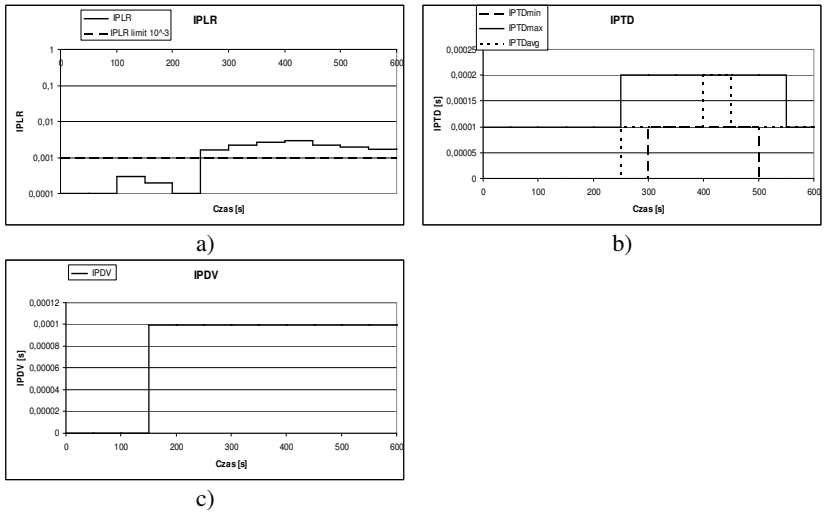



Fig.11 Scenario #3 – results; monitored link speed: 10Mbit/s; length of the buffer in access node: 10packets; a) IPLR, b) IPTD, c) IPDV

Presented results prove that our on-line monitoring method is suiTab.for fast detection of QoS level degradation that can follow from additional volume of load generated by users contrary to declared traffic contract. It probably would not be noticed when load was medium, but could be critical problem when links were fully utilized. Measurement in our method are performed on-line that guarantees short lag between incident in the

network and adequate information in results. In described example problem was observed after one measurement period since background traffic generator was started.

## 7. Summary

In the paper we presented how to perform on-line QoS monitoring of Real Time Class of Service in IP QoS networks. We assume that monitoring of the offered QoS level should be performed in worst-case traffic conditions allowed by the applied admission control algorithm. To maintain maximum allowed volume of traffic we introduce additional traffic to the network. The volume of this background traffic is a difference between maximum allowed traffic determined by CAC mechanism and payload.

Assumptions of the method were firstly verified during simulations than implemented and deployed in the testbed environment. We described details of the implementations issues and several simulation and practical experiments, which confirm the ability of our monitoring method for on-line monitoring of offered QoS of Real Time CoS.

# References


[1] The IST-EuQoS project, "End-to-end Quality of Service support over heterogeneous networks", www.ist-euqos.org

[2] The IST-DAIDALOS project, "Designing Advanced network Interfaces for the Delivery and Administration of Location independent**,** Optimised personal Services", www.ist-daidalos.org

[3] The GEANT project, www.geant.net

[4] The IST-MUSE project, "Multi Service Access Everywhere", www.ist-muse.org

[5] The IST-OPERA project, "Open PLC European Research Alliance", www.ist-opera.org

[6] W. Burakowski, M. Dąbrowski, M. Potts, EuQoS classes of Service, First European Workshop of End-to-End QoS in the Internet, Paris, June 2005

[7] J. Babiarz, K. Chan, F. Baker, "Configuration guidelines for DiffServ service classes", IETF draft-ietf-tsvwg-diffserv-service-classes-00, work in progress, February 2005

[8] ITU-T Recommendation Y.1541, "Network performance objectives for IP-based services", may 2002

[9] H. Tarasiuk, R. Janowski, W. Burakowski, "Admissible traffic load of real time class of service for inter-domain peers", In Proc. of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services, ICAS/ICNS 2005, published by IEEE Computer Society, 23-28 October 2005, Papeete, Tahiti, French Polynesia

[10] Xmlrpc web page, http://www.xmlrpc.com/spec

[11] Xmlrpc – Apache web page, http://ws.apache.org/xmlrpc/

[12] PostgreSQL web page, http://www.postgresql.org

[13] Reseaux IP European Network Coordination Center, http://www.ripe.net/

[14] F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, H. Uijterwaal, R. Wilhelm, "Providing Active Measurements as a Regular Service for ISP's", Passive and Active Measurements Workshop, PAM 2002

[15] C.J. Bovy, H.T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. Van Mieghem, "Analysis of End-to-end Delay Measurements in Internet", Passive and Active Measurements Workshop, PAM 2002

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 21–29

# Performance examination of the segment recovery in GMPLS

JANUSZ KORNIAK PAWEL ROZYCKI

University of Information Technology and Management
in Rzeszow
jkorniak@wsiz.rzeszow.pl, prozycki@wsiz.rzeszow.pl

**Abstract:** The GMPLS segment recovery method described in the new RFC recommendation promises rapid switch-over data traffic in the case of data plane failure. Authors present the NS-2 based simulations of segment recovery and shows first performance test with different topologies.
**Keywords:** : GMPLS, segment recovery, simulation

## 1. Introduction

The Generalized Multiprotocol Label Switching (GMPLS) [1] is considered as the next generation high-speed backbone network technology. In order to accommodate growing Internet traffic optical crossconnects (OCXs) must be implemented to reduce IP switching. The Automatically Switched Optical Networks (ASON) [2] and the GMPLS are key technology which satisfy the growing requirements. While these two concepts are developed separately in ITU-T and IETF the GMPLS can be considered as the implementation of the control plane for ASON.

The principles of the GMPLS network are:

- generalization of the label switching idea known from MPLS technique to all types of multiplexing,
- separation of functional planes:
  - **data plane** – includes all switching techniques supported by GMPLS such as WDM, TDM, packet switching etc.;
  - **common control plane** – IP network, called also signaling network, responsible for exchanging signaling and routing messages;
  - **management plane** – centralized or distributed supervise system that allows, for example, to employ provider's policy.

The signaling network of the GMPLS may be realized in general in the following manners:

- **in-band** – when signaling network is a part of data plane and signaling channels are implemented as logical channels separated from the supported transport technology (e.g. VC/VP within ATM or DCC within SDH/SONET);
- **out-of-band** – when signaling network id physically separated from the data plane.

The out-of-band signaling is important for implementing ASON technology. This approach allows avoiding the opto-electronic processing because data and control plane are physically separated. However such architecture can behave differently during in-band signaling. For example failure in data and control plane have to be maintained separately.

The out-of-band signaling allows to use different topologies for data and control planes. In this case two nodes directly connected in the data plane may be not connected in the control plane or, similarly, two nodes directly connected in the control plane may be not neighbors in the data plane. Such architecture of the GMPLS is called asymmetrical. If, however, the topologies of both planes are the same, the architecture is called symmetrical.

Many works are devoted the GMPLS reliability and especially the control plane reliability [3],[4]. The papers [5], [6] and [7] prepared by authors consider the some aspects of failure detection and notification, also for different types of architecture, in the GMPLS control plane when end-to-end protection of data plane is applied. The similar considerations for segment recovery mechanisms are presented in the next sections.

## 2. Segment recovery

The paper [8] defines the several types of protection for the GMPLS network including end-to-end protection and segment-based protection. In the end-to-end protection, described in details in the [9], the ingress and egress nodes for both working (called also protected) and backup (called protecting) LSP (Label Switched Path) are the same. This type of data plane protection can not satisfy the fast protection requirements when LSP is long. The propagation and processing delay can cause packet lost and higher delay. To solve this problem the concept of fast reroute [10] is introduced. For the GMPLS technology similar concepts is introduced in [11]. The node that initiates a recovery path is called *branch node* and the node that terminates such path is called *merge node*. The idea of the segment recovery is based on the protection of the segment LSP by other LSP called protection LSP. Note that each segment recovery LSP is established as an independent LSP and may be also protected using any protection method. The recovery LSP is established by the branch node. The path of recovery LSP may be defined by the ingress node of protected LSP or may be calculated by branch node himself. In the first case the path of given recovery LSP is carried by SERO (Secondary Explicit Route Object) object within the *Path* message. Note that the single *Path* message may includes multiple SERO objects, one for each defined segment recovery path. In the second case the path of segment recovery is calculated at each node along establishing

working LSP. The method of path calculation depends on content of SESSION_ATTRIBUTE object carried by *Path* message.

While the idea is simple and allows dividing LSP into multiple segments to satisfy fast protection switching, the signaling of the control plane becomes much more complicated. Signaling of segment recovery must coexist with the out-of-band signaling and support asymmetrical topology. Note that for given LSP one or more segment recovery LSP may be created and these paths may overlaps each other or may be nested within other segment recovery LSP.

This is important to note that the term *segment recovery* used in this section is the generalization of such mechanisms as segment protection and segment restoration. Described mechanism allows to use all well-known protection and restoration approaches such as 1+1 protection, 1:1 protection, LSP rerouting and shared-mesh restoration.

This new signaling requirements (segment recovery), out-of-band signaling and the asymmetrical topology can be a source of unpredicted behavior. In the next sections performance of GMPLS with segment recovery and mentioned assumptions is considered and verified by simulation.

## 3. Performance consideration

The reliability of GMPLS depends on many factors like:

- the data plane topology,
- the control plane topology,
- the recovery path method,
- the reliability of the control plane,
- the interaction between planes.

The first two items can be discussed in the context of planes symmetry or asymmetry. This problem was discussed in [5] for end-to-end protection method. The simulation presented in that wok shows dependency between topology asymmetry and reliability of GMPLS network. In the case of segment protection this influence is tested based on simulation shown in the next section.

The main aim of the segment recovery is to provide fast switch-over of data plane traffic and to satisfy low delay requirements. Therefore, only some recovery path methods can be used in this case. The recovery methods can be classified to computed on demand and pre-computed. The first method does not guarantee fast switching requirements. The second one can be classified to established on demand and pre-established. Only pre-established method can be used in segment recovery as a protection method which satisfy mentioned requirements. This method of path recovery is used in the simulations presented in the following sections.

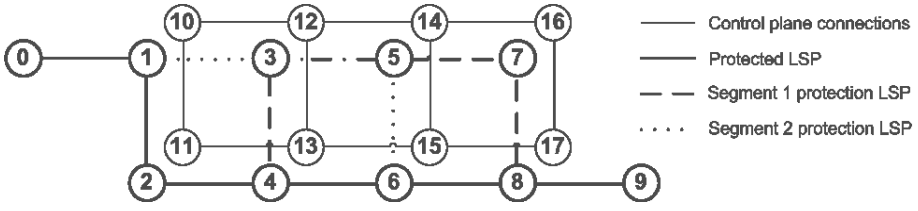



Fig. 1. The network topology used for tool validation

The reliability of the control plane is especially important in the case of separated functional planes and asymmetrical architecture. This problem was discussed in [5] for end-to-end protection method. It was shown that several enhancements in failure detection, notification and protection of the control plane significantly improve the overall reliability of GMPLS. For the segment recovery method similar signaling processing occurs and, therefore, proposed improvements should be verified.

The interaction between planes also can be factor of overall GMPLS reliability. For example high level of failures in data plane can be a source of high signaling load. Signaling packets can be processed with delay and negatively affects protection of data plane.

The next part of this paper focuses on the topology influence on the overall reliability when segment recovery is applied.

## 4. Performance examination of the segment recovery in GMPLS

The performance examination of the segment recovery in the GMPLS can be performed by simulation. This approach allows for quantity measurement of technology performance. The NS-2 [12] - network simulator has been used to conduct our research.

### 4.1. Simulation tool - verification and validation

The NS-2 is not a GMPLS simulator however it is possible to extend this environment to support simulation of this technology [13]. The base of our simulator is the MPLS tool included in NS-2 and the RSVP-TE implementation provided by [14]. In order to adopt this simulator to our task the following expansions have been prepared:

- physical separation of control and data planes,
- supporting for asymmetrical architecture,
- end-to-end data plane protection methods,
- various recovery strategies

The architecture of implemented extensions are described in [15]. The newly added functionality is the segment recovery method. This functionality must be tested and verified.

Therefore, several simple simulation has been prepared. Simple network used for validation of segment protection method is depicted on figure 1.

The tests performed for verification includes:

- establishment of segment protection LSP
- segment recovery (switch over)
- control plane signaling after failure
- release of reservation for protection LSP

Figure 2 illustrates several steps of testing process. The first step is the LSP establishment. In this step the protected LSP is created by sending *Path* message from ingress to egress. This message contain *SERO* object including planed protection LSP. The protection LSP establishment is initiated on *branch* node not before *Resv* message comes back from egress node. Thus branch node send the *Path* message to *Merge* point end expect return *Resv* massage send by *Merge*. Established protection LSP is independent one with special *Association* object which is used to assign with protected LSP. Processing of protected and protection LSP establishment is designed according RFC recommendation [11].

The key step of verification is the test of protection by segment recovery. The *PathErr* message processing was modified according to RFC recommendation to support switch-over to protection LSP. Therefore, after a failure in the data plane the *PathErr* is send upstream. Typically, this message is processed up to the ingress but in the segment recovery is processed to the first *branch* point which can support protection of failed segment. After the protection LSP is established Forward Information Base (FIB) contains necessary information for switching data from protected LSP on all nodes except *Branche* point. When the *PathErr* is received on *Branche* node the FIB is modified and data is switched-over immediately.

The last step of test is the release process. The *PathTear* messages are processed in protected LSP but also must by pushed to protection LSP. Also when protected LSP fails and protection is in use *PathTear* must be forwarded from protection to protected LSP.

### 4.2. Model of GMPLS network and performance test

The topology used for performance test is depicted on figure 3. Both the control and data plane are symmetrical. The source of data flow (UDP stream) are nodes 0, 6, 12 and 18. The destination is chosen from the same set of nodes. Thus there are four pair of source and destination. The pair of source and destination, start time and duration of flow is chosen randomly. Moreover, random number on randomly chosen links are planed to fail on data plane. The Label Switched Paths pre-established and pre-allocated based on routing information (OSPF-TE). In order to measure the performance of service number of lost packet is monitored.

Several simulations are started with different protection method and with symmetrical and asymmetrical topology. The first simulation is used as a baseline. In this simulation data plane is not protected. Table 1 shows the results of this simulation.

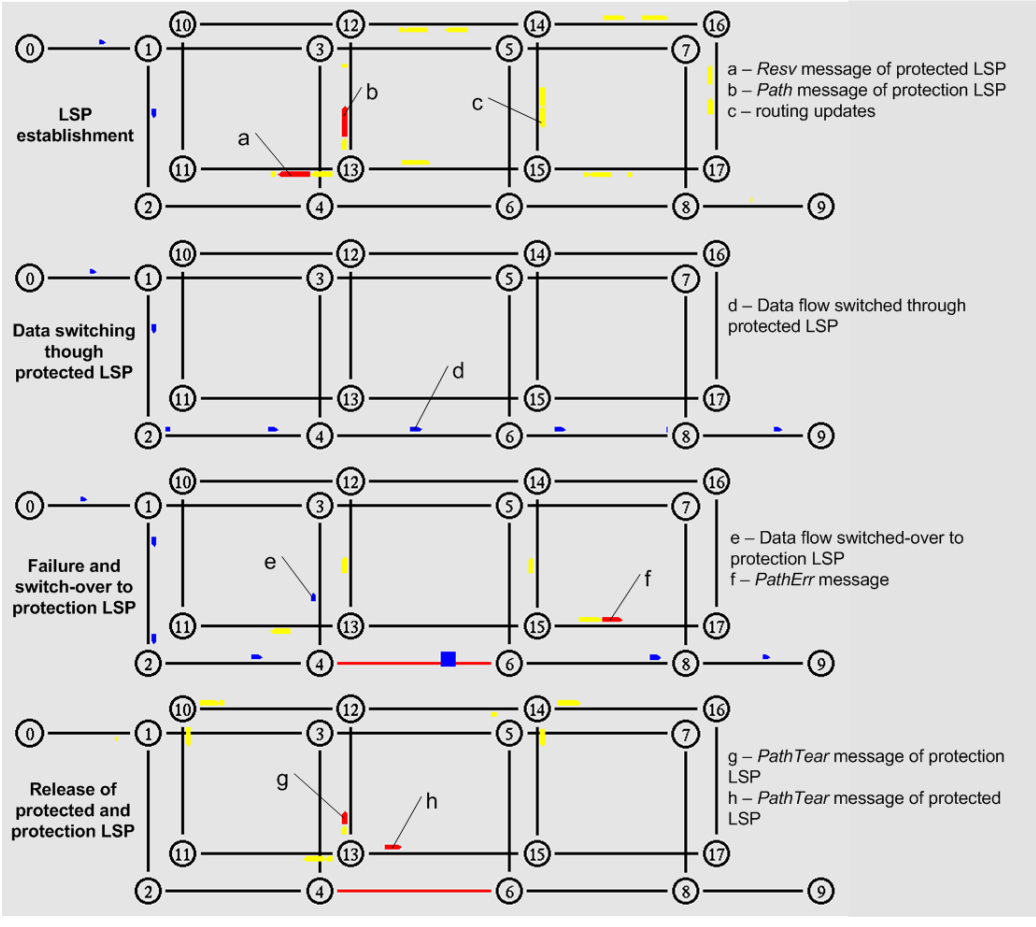



Fig. 2. Verification of simulation module for segment recovery.

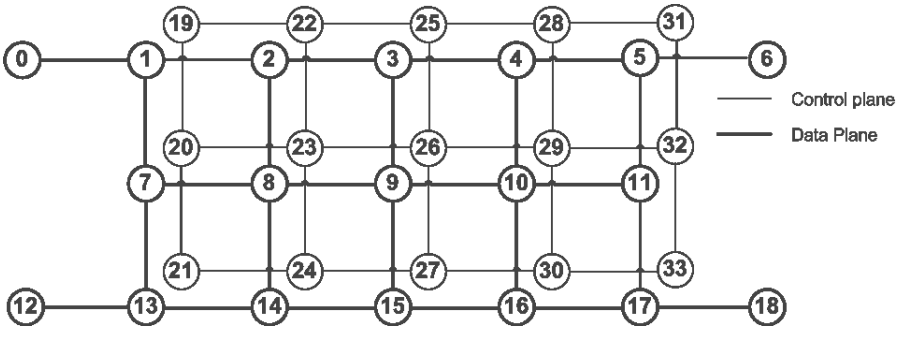



Fig. 3. Network topology used for simulation

The second simulation uses segment recovery technique and symmetrical topology. Table 2 presents the simulation results.

The next simulation with segment recovery and asymmetrical topology takes the same results in spite of disabled 6 links in control plane. The last two simulation uses the end-to-end protection with symmetrical and asymmetrical topology. The number of lost packets is very high (350 to 1000).

Table 1. Simulation without protection

| | |
|---|---|
| Simulation time | 10s |
| Number of LSP | 10 |
| Number of packed send | 4681 |
| Number of failures | 5 |
| Number of packet lost | 1106 |

Table 2. Simulation with segment recovery

| | |
|---|---|
| Simulation time | 10s |
| Number of LSP | 10 |
| Number of packed send | 4681 |
| Number of failures | 5 |
| Number of packet lost | 4 |

### 4.3. Analysis of the results

In the second and the third simulation the number of lost packets is very low as suspected. The recovery LSP are short and overlaps. Therefore primary LSP is protected in spite of failure in one recovery LSP. The asymmetry of the functional planes has no great impact to the performance. Again, due to short recovery paths. In the last two simulations the number of lost packets reaches almost 1000. This is the result of failure in protection LSP. The protection LSP are long and, therefore, end-to-end protection is susceptible for failures. The number of link failure in data plane is five within 10 seconds. It almost completely destroys protection of primary LSPs.

## 5. Conclusions

The simulation results confirm the efficiency of the segment recovery technique. Especially when the size of protected segment is not too big. Switching to recovery LSP is almost immediate. In our simulation only one packet is lost per failure. Moreover overlapping recovery LSP protected certain LSP is very effective. If one recovery LSP is failed other can

still support recovery for failed link or node. The end-to-end protection is much less efficient if used without any method for backup LSP reestablishment. The disadvantage of recovery LSP is high level of bandwidth reservation. Especially when recovery LSP are overlapping. However the size of the segment can be adjusted to satisfy low delay requirement and cost of resources reservation.

The influence of the asymmetry on the overall performance is insignificant when protection LSP is short and increase when protection LSP becomes longer.

The in-depth analysis of efficiency of the segment recovery mechanisms require much more well prepared simulations. The future work will be, therefore, focused on detailed study of influence of described mechanism on such parameters as MTTR (*Mean Time To Restore*) and intensity of the control plane messages taking into consideration different topologies and architectures of the functional planes.

## References

[1] E.Mannie (Ed.) et al.: Generalized Multi-Protocol Label Switching (GMPLS) Architecture, *RFC 3945* October 2004.

[2] ITU-T Rec. G.8080/Y1304: Architecture for the Automatically Switched Optical Network (ASON), Nov. 2001 (rev. Jan. 2003).

[3] G. Li, J. Yates, D. Wang, C. Kalmanek: Control plane design for reliable optical networks, *IEEE Communication Magazine*, pp. 90-96, February 2002.

[4] A. Jajszczyk, P. Rozycki: Recovery of the Control Plane after Failures in ASON/GMPLS Networks, *IEEE Network*, vol.20 No.1, Jan/Feb 2006.

[5] P. Rozycki, J. Korniak, A. Jajszczyk: Failure Detection and Notification in GMPLS Control Plane, *IEEE ICC 2007*, "Workshop on GMPLS Performance Evaluation: Control Plane Resilience", 24 June 2007, Glasgow.

[6] J. Korniak, P. Rozycki “Signaling improvements for GMPLS control plane”, Tools of the Information Technology 2007, Rzeszow, (accepted)

[7] P. Rozycki, J. Korniak “End-to-end protection strategies in the GMPLS networks”, Tools of the Information Technology 2007, Rzeszow, (accepted)

[8] E. Mannie, D. Papadimitriou (Ed.) et al.: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS), *RFC4427*, March 2006

[9] J.P. Lang, Y. Rekhter, D. Papadimitriou, (Ed.) et al.: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery, *RFC4872*, May 2007

[10] P. Pan, G. Swallow, A. Atlas: Fast Reroute Extensions to RSVP-TE for LSP Tunnels, *RFC 4090* May, 2005.

[11] L. Berger, I. Bryskin, D. Papadimitriou, A. Farrel: GMPLS Segment Recovery, *RFC 4873* May 2007.

[12] VINT project at LBL, Xerox PARC, USB and UCS/ISI The Network Simulator ns-2 http://www.isi.edu/nsnam/ns/

[13] J. Korniak, P. Rozycki, "GMPLS – simulation tools", *Proceedings of the 1st Conference "Tools of Information Technology"*, Rzeszow, Poland, 2006

[14] C. Callegari, F.Vitucci, RSVP-TE patch for MNS/ns-2 http:// netgroup-serv.iet.unipi.it/rsvp-te_ns/

[15] J. Korniak, P. Rozycki, Modelowanie sieci GMPLS w srodowisku NS-2, XIV Konferencja Sieci komputerowe, Zakopane 2007, ISBN 978-83-206-1649-1 - In Polish

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 31 - 42

# Quality of service in optical burst switched networks

SYLWESTER KACZMAREK [a] PAWEŁ KOSTECKI

Faculty ETI, Department STI
Gdańsk University of Technology
[a] *Sylwester.Kaczmarek@eti.pg.gda.pl*

**Abstract:** In the paper analytical models of two service differentiation schemes for optical burst switched network: extended offset time based and PPS (Preemptive Priority Scheme) are revised. Also accordance of analytical models for those schemes is studied when complete class isolation is assumed. Furthermore authors introduce an analytical model which describes an effective degree of isolation when burst switched network employs both schemes and JET signaling. The comparison of analytical and simulation results are also included.
**Keywords:** OBS, QoS, offset time, PPS, JET signaling, analytical and simulation model

## 1. Introduction

Optical burst switching is considered as a promising solution for all-optical next generation network. An important issue in optical burst switched networks is service differentiation. So far many proposals have been published within this subject. The most significant service differentiation schemes are based on extended offset time [1], preemptive priorities PPS [6] and PPBS [3] or intentional burst dropping (proportional QoS [4]).

In extended offset time based scheme lower burst loss probability for a certain class can be achieved by extending its offset time. If the offset time difference between class *i* and class *j* is large enough, then class *i* is completely isolated from class *j*, which means that class *i* burst cannot be blocked by class *j* burst. The analytical model for extended offset time scheme is presented in [1][2] and it describes burst loss probability in a single node.

In PPS it is assumed that higher priority burst can preempt lower priority burst with a certain probability when lower priority burst is blocking higher priority burst. Analytical model that describes burst loss probability in a single node is presented in [6].

Service differentiation by extending offset time comes off natural properties of signaling for distributed control architecture in optical burst switched network. Therefore if a PPS scheme is used along with JET signaling [7] both mentioned service differentiation scheme could be applied.

When both schemes are applied the effective degree of class isolation depends on offset time difference and preemption probability. Different values of offset time

difference and preemption probability can result in the same value of effective degree of isolation. Therefore the same value of loss probability can be achieved for different values of those parameters. Analytical model for calculating effective degree of isolation will be introduced and validated by simulation.

When both schemes assume complete class isolation then analytical models, even though they are different, should give the same loss probabilities. This also will be verified mathematically and by simulation.

The paper is organized as follows. In section 2 analytical models for extended offset time scheme and PPS are reminded and also a model for combination of those two schemes is introduced. Also in this section the simulation model is described. The results of research are presented and discussed in section 3 and the paper is ended with the conclusions.

## 2. Analytical and simulation models

### 2.1. Extended offset time

Analytical model for extended offset time based scheme [2] assumes complete class isolation, full wavelength conversion, non-blocking switch and $N$ wavelengths in outgoing link.

If there are $K$ class of traffic (0…$K$-1, where 0 means the highest priority) and class $i$ ($i$ = 0, 1, …, $K$ – 1) is completely isolated from any other lower priority class then burst loss probability for class $i$ can be calculated from:

$$B_i = \frac{\left(\sum_{j=0}^{i} \lambda_j\right) B_{0,i} - \left(\sum_{j=0}^{i-1} \lambda_j\right) B_{0,i-1}}{\lambda_i}, \qquad (1)$$

where $\lambda_i$ is class $i$ burst arrival rate and $B_{0,i}$ is burst loss probability averaged over class 0 through class $i$. Averaged burst loss probability $B_{0,i}$ can be obtained from Erlang-B formula:

$$B_{0,i} = E_{1,N}\left(\sum_{j=0}^{i} A_j\right), \qquad (2)$$

where $A_j$ is the offered load for class $j$.

The degree of isolation $R_{i,j}$ was also introduced in [1]. It is defined as a probability that class $j$ burst cannot block class $i$ burst. Its value can be calculated from following equation:

$$R_{i,j} = 1 - P(t_{off}{}^{i} - t_{off}{}^{j} > \underline{L}_j), \qquad (3)$$

where $t_{off}^{i}$ is the offset time for class $i$ and $L_j$ is class $j$ burst duration.

Presented analytical model is only accurate if $R_{i,j}$ = 1 as it assumes complete class isolation.

## 2.2. Preemptive Priority Scheme

Analytical model for PPS service differentiation scheme [6] assumes only two class of traffic (namely 0 and 1, where 0 means highest priority), full wavelength conversion, non-blocking switch and $N$ wavelengths in outgoing link. It is based on multi-dimensional Markov chain. In PPS class 0 burst can preempt class 1 burst with probability $p_{0,1}$ when class 1 burst is blocking class 0 burst. Burst loss probability for class 0 can be calculated from:

$$B_0 = P_{N,0} + (1 - p_{0,1}) \sum_{\substack{m+n=N \\ m<N}} P_{m,n} , \qquad (4)$$

where $P_{m,n}$ is the probability of state in which service system is serving $m$ bursts of class 0 and $n$ bursts of class 1. Burst loss probability for class 1 can be obtained from:

$$B_1 = \sum_{m+n=N} P_{m,n} + \frac{\lambda_0}{\lambda_1} p_{0,1} \sum_{\substack{m+n=N \\ n>0}} P_{m,n} . \qquad (5)$$

The probabilities $P_{m,n}$ are calculated from system steady state equations. Steady state equations can be obtained from system state transition diagrams shown in Fig. 1.

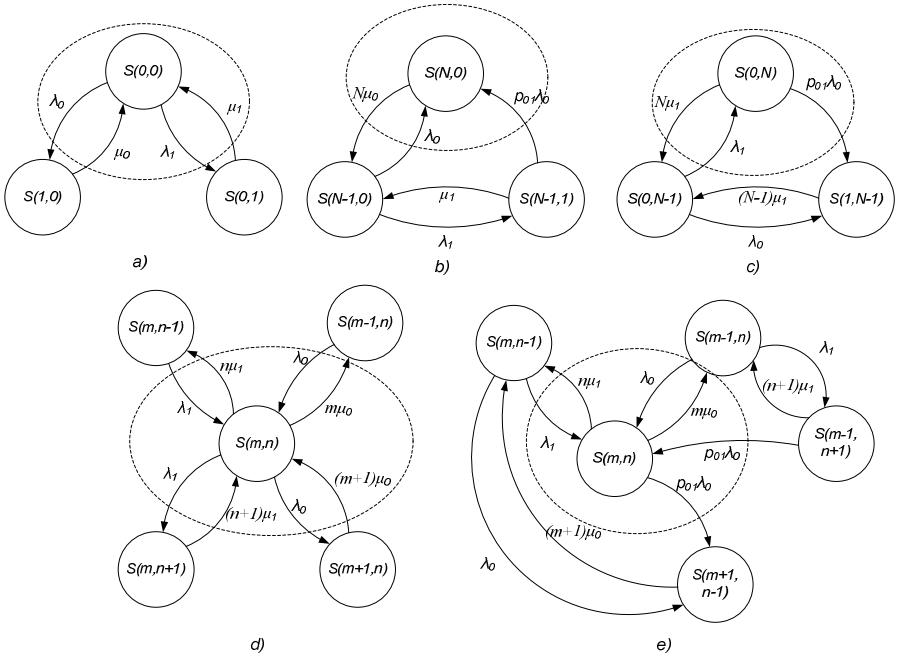



Fig. 1. State transition diagrams for certain cases; *S(m,n)* stands for state in which system is serving $m$ class 0 bursts and $n$ class 1 bursts; a) *S(0,0)*; b) *S(N,0)*; c) *S(0,N)*; d) *S(m,n)*, where: $0 < m + n < N$; e) *S(m,n)*, where: $m + n = N$, $m < N$, $n < N$.

Based on Fig. 1 system steady state equations can be written as follows:

$$\begin{cases} 0 = -P_{0,0}(\lambda_0 + \lambda_1) + \mu_0 P_{1,0} + \mu_1 P_{0,1} \\ 0 = -P_{m,n}(\lambda_0 + \lambda_1 + m\mu_0 + n\lambda_1) + P_{m-1,n}\lambda_0 + P_{m,n-1}\lambda_1 + \\ + P_{m+1,n}(m+1)\mu_0 + P_{m,n+1}(n+1)\mu_1 \ , 0 < m+n < N \\ 0 = -P_{m,n}(p_{0,1}\lambda_0 + m\mu_1 + n\mu_0) + P_{m-1,n+1}p_{0,1}\lambda_0 + \\ + P_{m-1,n}\lambda_0 + P_{m,n-1}\lambda_1 \ , m+n = N, m < N, n < N \\ 0 = -P_{N,0}N\mu_0 + P_{N-1,0}\lambda_0 + P_{N-1,1}p_{0,1}\lambda_0 \\ 0 = -P_{0,N}(p_{0,1}\lambda_0 + N\mu_1) + P_{0,N-1}\lambda_1 \end{cases} \quad (6)$$

Analytical model for PPS doesn't take into account that segmentation of low priority bursts can occur and therefore class 1 loss probability might be lower. Segmentation is only possible when offset time for class 0 is shorten then class 1 burst duration and it happens when both class 0 burst and it's header arrive when class 1 burst is being serviced (Fig. 2).

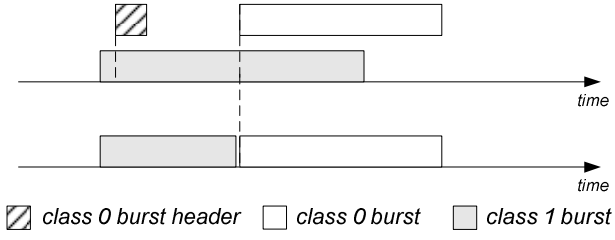



Fig. 2. Burst segmentation in PPS.

### 2.3. Combination of both models

If both schemes are applied and only two class of service are assumed then the degree of isolation results from $R_{0,1}$ and $p_{0,1}$.

Lets assume situation when class 0 burst arrives in state *S(N-n,n)* in which *N-n* class 0 bursts and $n$ ($n > 0$) class 1 bursts are being serviced and the $N$ is the number of wavelengths in outgoing link.

Newly incoming class 0 burst in state *S(N-n,n)* can be served without preempting any of $n$ class 1 bursts if the offset time difference is large enough that class 0 burst won't see at least one class 1 reservation. For a single wavelength probability that class 0 burst won't see class 1 reservation is equal to $R_{0,1}$. If there are $n$ class 1 reservations and bursts are served independently then that probability is equal to $1-(1-R_{0,1})^n$.

On the other hand preemption will occur if newly incoming class 0 burst sees all reservations made for class 1 bursts. Therefore preemption in state *S(N-n,n)* occurs with probability $p_{0,1}(1-R_{0,1})^n$.

Probability that newly incoming class 0 burst in state *S(N-n,n)* is successfully served can be written as follows:

$$p_{0,1}(S(N-n,n)) = p_{0,1}\left(1-R_{0,1}\right)^n + 1 - (1-R_{0,1})^n . \quad (7)$$

Equation (7) defines the effective degree of isolation in state *S(N-n,n)*.

Analytical model for PPS can be adopted to obtain burst loss probabilities for combination of both models by replacing $p_{0,1}$ by $p_{0,1}(S(N-n,n))$. Therefore system steady state equations (6) can be rewritten as follows:

$$\begin{cases} 0 = -P_{0,0}(\lambda_0 + \lambda_1) + \mu_0 P_{1,0} + \mu_1 P_{0,1} \\ 0 = -P_{m,n}(\lambda_0 + \lambda_1 + m\mu_0 + n\lambda_1) + P_{m-1,n}\lambda_0 + P_{m,n-1}\lambda_1 + \\ + P_{m+1,n}(m+1)\mu_0 + P_{m,n+1}(n+1)\mu_1 \ , 0 < m+n < N \\ 0 = -P_{m,n}(p_{0,1}(S(m,n))\lambda_0 + m\mu_1 + n\mu_0) + P_{m-1,n+1}p_{0,1}(S(m-1,n+1))\lambda_0 + \\ + P_{m-1,n}\lambda_0 + P_{m,n-1}\lambda_1 \ , m+n = N, m < N, n < N \\ 0 = -P_{N,0}N\mu_0 + P_{N-1,0}\lambda_0 + P_{N-1,1}p_{0,1}(S(N-1,1))\lambda_0 \\ 0 = -P_{0,N}(p_{0,1}(S(0,N))\lambda_0 + N\mu_1) + P_{0,N-1}\lambda_1 \end{cases}, \quad (8)$$

and burst loss probabilities can be calculated from following equations:

$$B_0 = P_{N,0} + (1 - p_{0,1}(S(m,n))) \sum_{\substack{m+n=N \\ m<N}} P_{m,n} \ , \quad (9)$$

$$B_1 = \sum_{m+n=N} P_{m,n} + \frac{\lambda_0}{\lambda_1} p_{0,1}(S(m,n)) \sum_{\substack{m+n=N \\ n>0}} P_{m,n} \ . \quad (10)$$

In Fig. 3 and Fig. 4 we show the effective degree of isolation as a function of $p_{0,1}$ and $R_{0,1}$ for *S(N,0)* and *S(N-1, 1)* when $N = 8$ and offered load is 0.8Erl per wavelength.

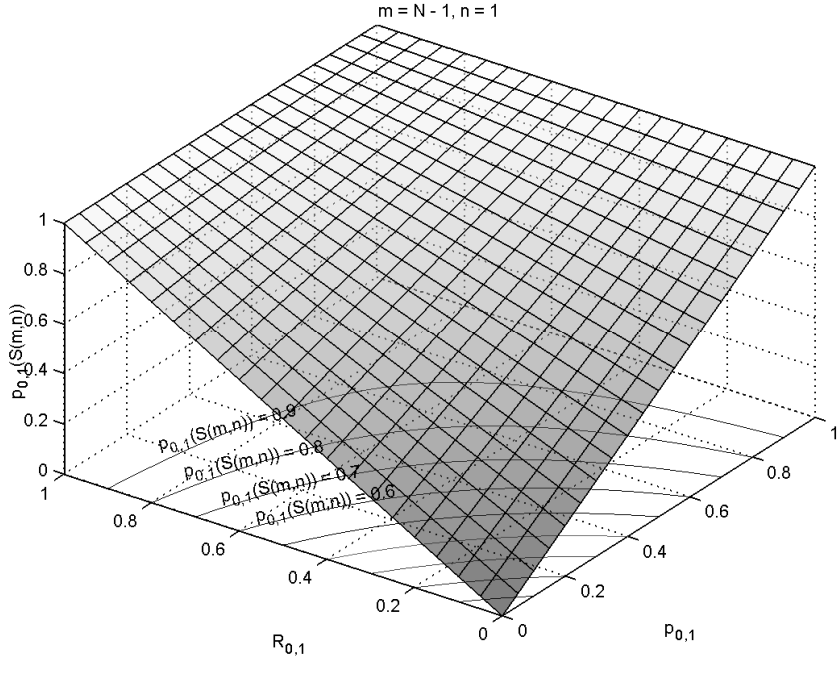



Fig. 3. Effective degree of isolation as a function of $p_{0,1}$ and $R_{0,1}$ for *S(N-1,1)* when $N = 8$.

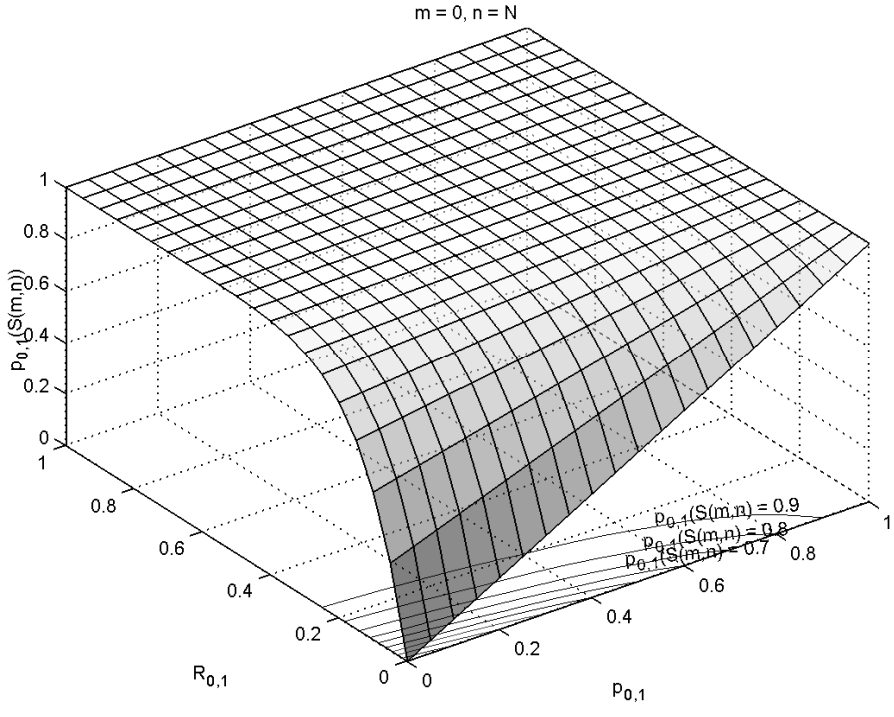



Fig. 4. Effective degree of isolation as a function of $p_{0,1}$ and $R_{0,1}$ for $S(0,N)$ when $N = 8$.

It can be concluded from Fig. 3 and Fig. 4 that the same value of effective degree of isolation in certain state $S(N-n,n)$ can be achieved for different $p_{0,1}$ and $R_{0,1}$ values.

Moreover analytical model for PPS should lead to the same results as model for extended offset time based scheme in case of complete class isolation (i.e. when $R_{0,1} = 1$ and $p_{0,1} = 1$). Consideration of burst loss probabilities under assumption of complete class isolation gives the following results (based on equations (1), (2) for offset time based scheme and (4), (5), (6) for PPS):

$$B_{0(PPS)} = \frac{A_0}{1 + A_0}, \tag{8}$$

$$B_{1(PPS)} = \frac{A_0 + A_1 + A_0 \frac{\lambda_0}{\mu_1} + A_1 \frac{\lambda_0}{\lambda_1} + A_0 A_1}{\left(1 + A_1 + \frac{\lambda_0}{\mu_1}\right) \cdot \left(1 + A_0\right)}, \tag{9}$$

$$B_{0(offset)} = \frac{A_0}{1 + A_0}, \tag{10}$$

$$B_{1(offset)} = \frac{A_0 + A_1 + A_0^{\,2} + A_1 \dfrac{\lambda_0}{\lambda_1} + A_0 A_1}{\left(1 + A_1 + A_0\right) \cdot \left(1 + A_0\right)}, \qquad (11)$$

where $\mu_i$ is the class $i$ service rate. Burst loss probabilities for class 0 are equal, but loss probabilities for class 1 differ. In fact burst loss probabilities for class 1 are different unless service rates for class 0 and class 1 are equal ($\mu_0 = \mu_1$).

### 2.4. Simulation model

For the need of the analytical models verification an event-driven burst switched network simulator has been developed. Simulator is completely written in JAVA language and it takes advantage of object oriented programming.

Simulator implements both service differentiation schemes and JET signaling. For extended offset time based scheme an LAUC-VF channel scheduling algorithm [5] is provided, while PPS is using modified version of LAUC-VF algorithm (it takes priorities into account).

Channel scheduling algorithm for PPS looks for an idle channel in a LAUC-VF manner. If there is no idle channel then channel occupied by lower priority burst and with minimum overlapping is chosen. Therefore newly incoming burst is served when:

- there is an idle channel,
- or there are no idle channels and at least one lower priority burst is being serviced and preemption occurs.

Analytical models are validated by simulation assuming two class of traffic. For extended offset time based scheme and PPS, burst loss probability is measured in a function of total offered load per wavelength. In both cases complete class isolation is assumed (i.e. $R_{0,1} = 1$ and $p_{0,1} = 1$). Bursts lengths for class 0 and 1 are exponentially distributed with mean 100 kB. Transmission rate on a single wavelength is 1 Gb/s. Burst arrival rate has also exponential distribution and its mean value is adjusted to achieve proper offer load value. Burst arrival rates are equal for both classes ($\lambda_0 = \lambda_1$). For PPS, the offset time for class 0 is much longer then mean burst duration of class 1 so burst segmentation won't happen.

When both schemes are applied, two cases are simulated: when ($p_{0,1} = 0.5$, $R_{0,1} = 0.75$) and ($p_{0,1} = 0.75$, $R_{0,1} = 0.5$). Other simulation conditions are same as in previous simulations. Simulation results are compared with analytical one for superposition of both schemes.

Finally extended offset time based scheme is simulated when service rates for class 0 and class 1 are different and $R_{0,1} = 1$. Two cases are studied: when $\mu_0 = 0.25\mu_1$ and $\mu_0 = 4\mu_1$. Burst arrival rates are equal for class 0 and 1 ($\lambda_0 = \lambda_1$) and are adjusted to achieve proper value of offered load. Simulation results are compared with analytical one for both schemes.

The results are presented in the following section. Simulation results are presented with 95% confidence interval.

## 3. Results

From Fig. 5 and Fig. 6 it can be concluded that analytical models for extended offset time scheme and PPS are quite accurate. In both cases when $N = 1$ loss probability is smaller then in theory, but when $N = 4$ or $N = 8$ there is no significant difference between analytical and simulation results.

For superposition of both schemes (Fig. 7 and Fig. 8) it can observed that introduced analytical model is accurate. It can also be concluded that some values of $p_{0,1}$ and $R_{0,1}$ result in similar burst loss probabilities. In both cases (($p_{0,1} = 0.5$, $R_{0,1} = 0.75$), ($p_{0,1} = 0.75$, $R_{0,1} = 0.5$)) class 1 burst loss probabilities are nearly equal. Class 0 burst loss probabilities differ, but this difference is marginal.

When service rates for class 0 and 1 are different (Fig. 9 and Fig. 10) analytical model for extended offset time scheme gives lesser burst loss probability for class 1 when $\mu_0 = 0.25\mu_1$ and higher when $\mu_0 = 4\mu_1$. Analytical model for PPS is more accurate however analytical model for offset time based scheme can be easily adopted to calculate loss probabilities when there are more than two service class. In PPS it seems to be much more difficult.

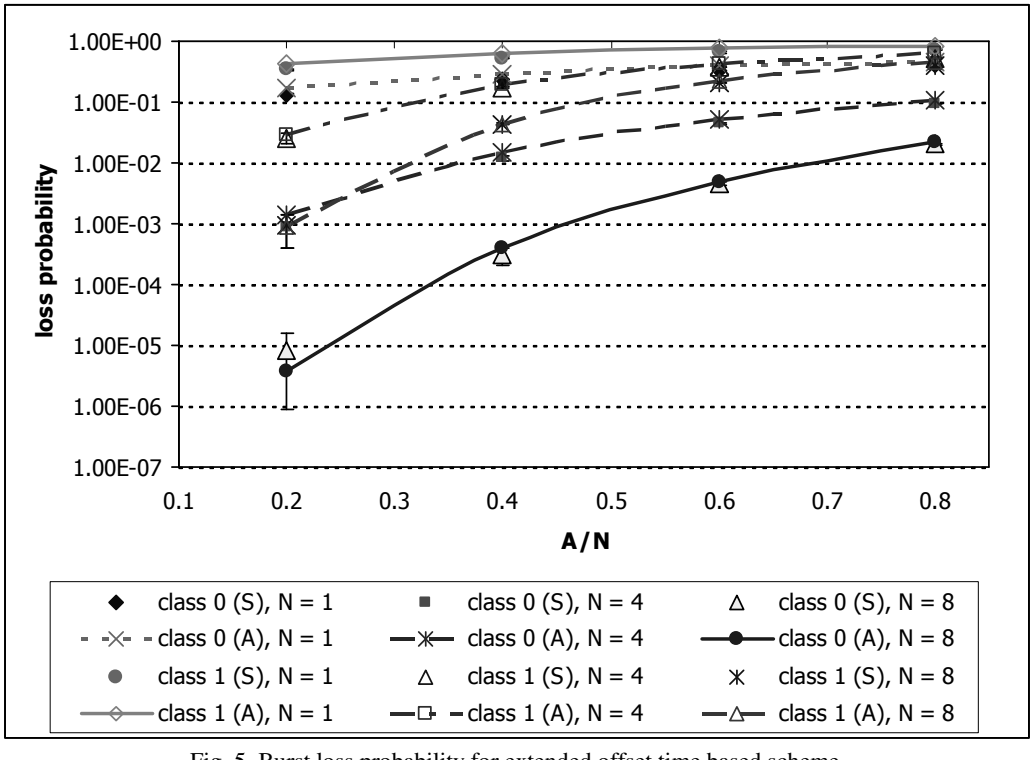



Fig. 5. Burst loss probability for extended offset time based scheme.

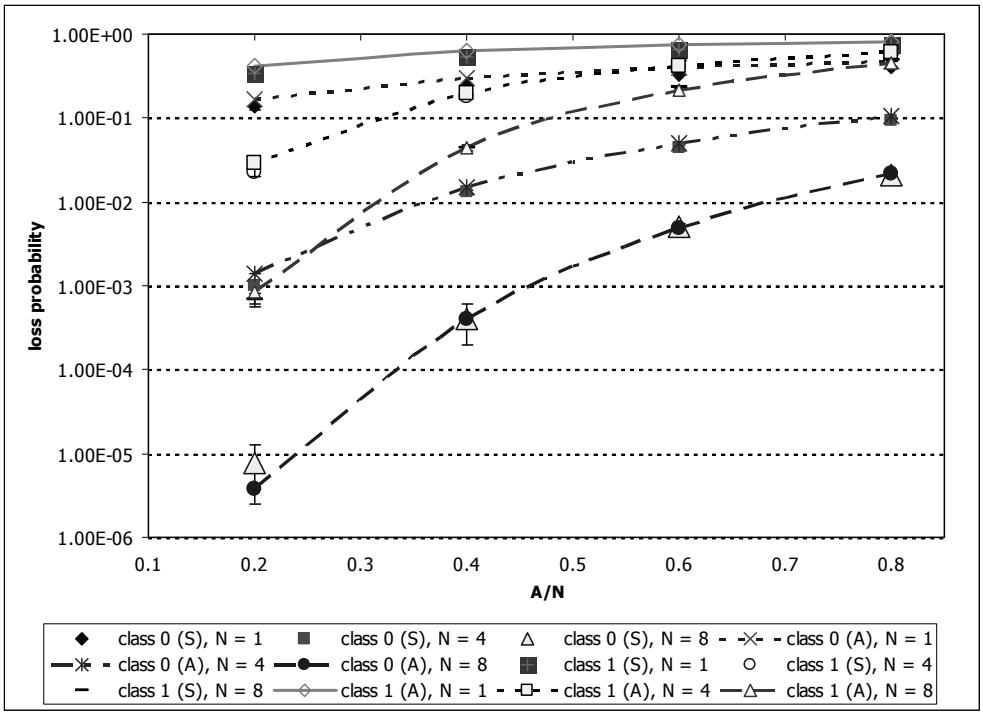



Fig. 6. Burst loss probability for PPS when $p_{0,1} = 1$.

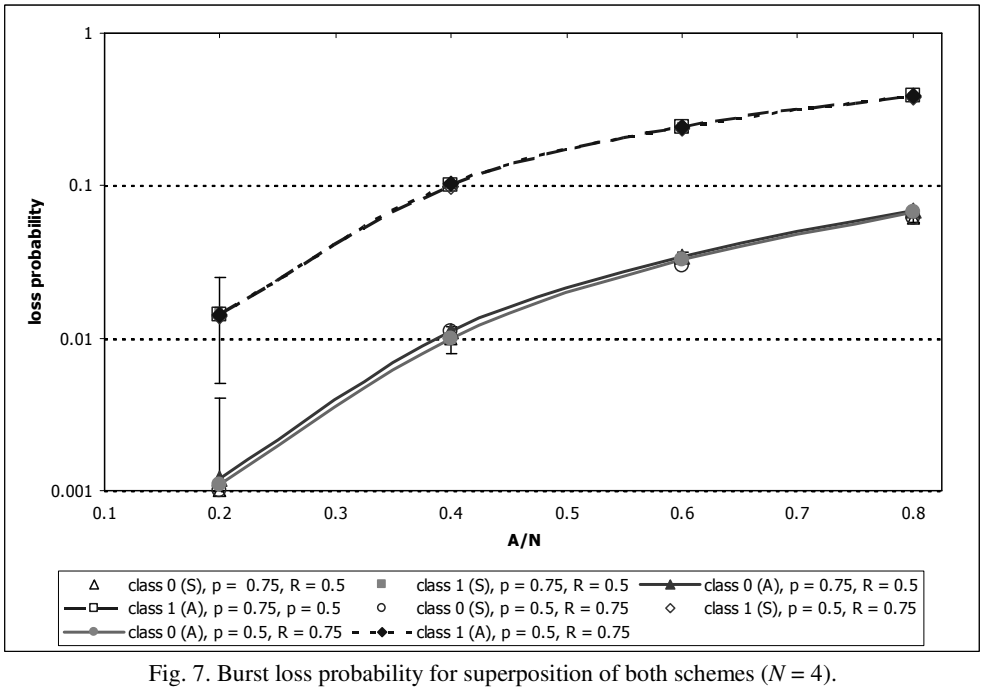



Fig. 7. Burst loss probability for superposition of both schemes ($N = 4$).

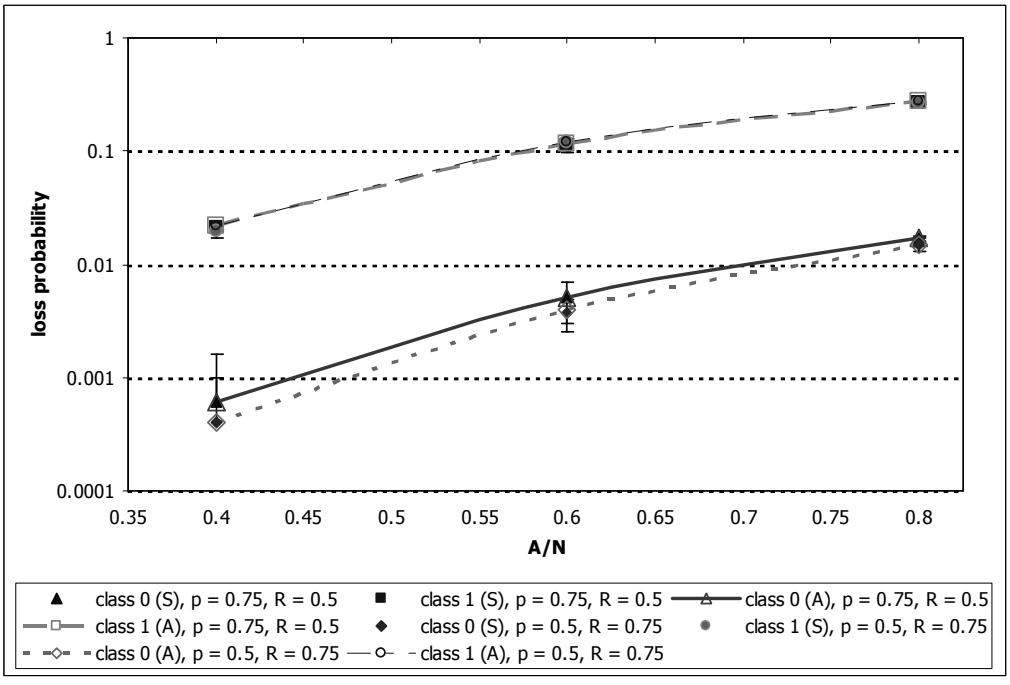



Fig. 8. Burst loss probability for superposition of both schemes ($N = 8$).

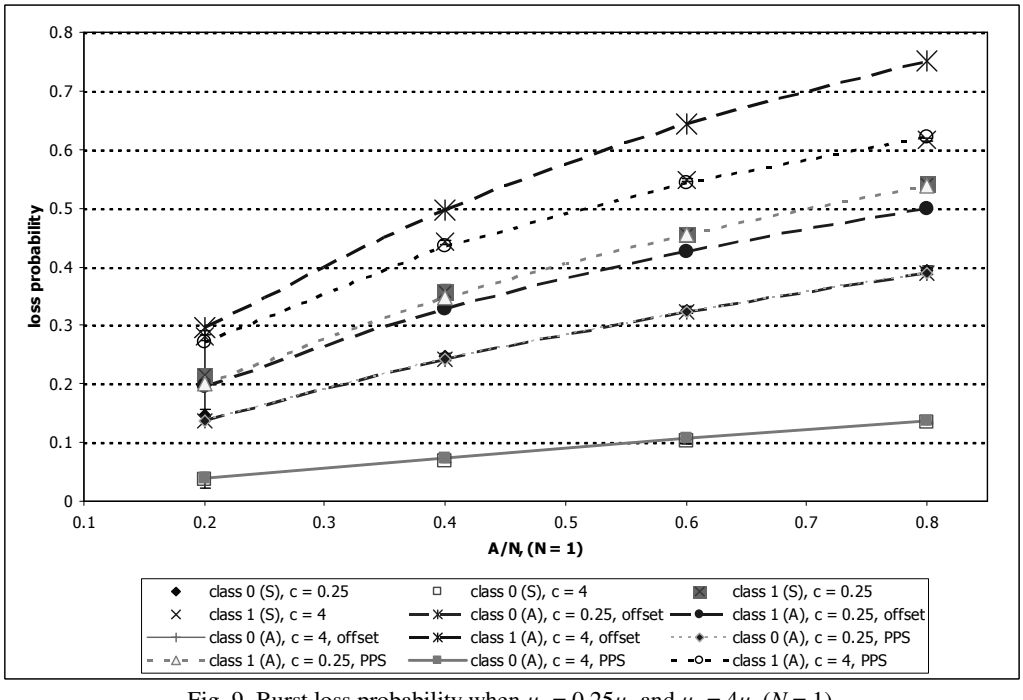



Fig. 9. Burst loss probability when $\mu_0 = 0.25\mu_1$ and $\mu_0 = 4\mu_1$ ($N = 1$).

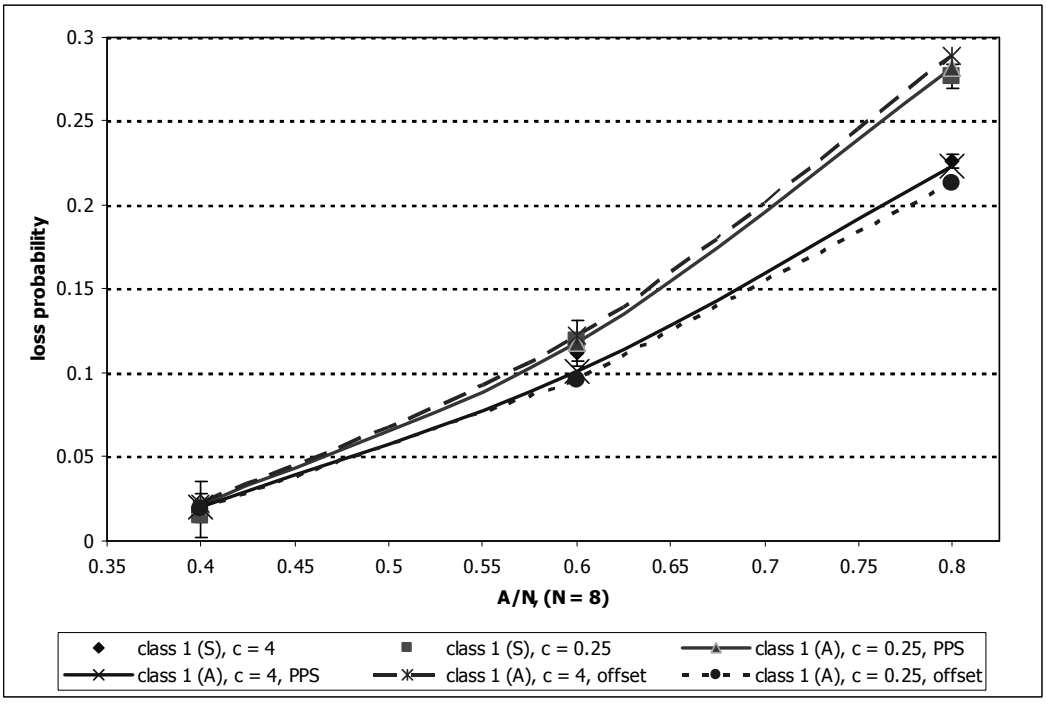



Fig. 10. Burst loss probability when $\mu_0 = 0.25\mu_1$ and $\mu_0 = 4\mu_1$ ($N = 8$, only class 1 results presented for readability).

## 4. Concluding remarks

Analytical models for extended offset time based scheme and PPS are quite accurate. Analytical model for PPS can be adopted to obtain burst loss probability even when extended offset time scheme is applied and it is more accurate then the original one. From simulation results when both schemes were applied it can be concluded that introduced analytical model can be used to accurately calculate burst loss probability.

Presented service differentiation schemes offer acceptable burst loss probability in a single node under typical load (i.e. $A/N$ ~ 0.6Erl – 0.8Erl) if number of wavelengths is relatively large (for example $N = 128$) and that probability can be estimated by analytical models. Fiber delay lines can be used to reduce loss probability as was shown in [1], but today adding FDL may dramatically increase cost of a single node.

# References

[1] Yoo M., Qiao C., Dixit S.: *QoS Performance of Optical Burst Switching in IP-Over-WDM Networks*. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, Oct. 2000, p. 2062 – 2071.

[2] Zukerman M., Vu H. L.: *Blocking Probability for Priority Classes in Optical Burst Switching Networks*. IEEE Communications Letters, Vol. 6, No. 5, May 2002, p. 214 – 216.

[3] Tan C. W., Mohan G, Lui J. C.: *Achieving Multi-Class Service Differentiation in WDM Optical Burst Switching Networks: A Probabilistic Preemptive Burst Segmentation Scheme*. IEEE Journal on Selected Areas in Communications, Vol. 24, No. 12, December 2006, p. 106 – 119.

[4] Chen Y., Hamdi M., and Tsang D. H. K.: *Proportional QoS over OBS network*. *IEEE GLOBECOM*, vol. 3, San Antonio, TX, Nov. 2001, p. 1510–1514.

[5] Yijun Xiong, Vandenhoute M., Cankaya H.C.: *Control Architecture in optical burst-witched WDM networks*. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, Oct. 2000, p. 1838 – 1851.

[6] Yang L., Jiang Y., Jiang S.: *A Probabilistic Preemptive Scheme for Providing Service Differentiation in OBS Networks*. IEEE GLOBECOM 2003, pp. 2689 – 2693.

[7] Qiao C., Yoo M.: *Optical Burst Switching (OBS)- A New Paradigm for an Optical Internet*. J. High Speed Nets., vol. 8, no. 1, Jan. 1999, pp. 69 - 84.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 43 - 51

# New switch architectures with optical buffering using QD-SOA devices

WOJCIECH KABACIŃSKI [a] SŁAWOMIR WĘCLEWSKI [b]

[a] Chair of Telecommunication and Computer Networks
Poznan University of Technology
*wojciech.kabacinski@et.put.poznan.pl*

[b] Chair of Telecommunication and Computer Networks
Poznan University of Technology
*sweclew@et.put.poznan.pl*

**Abstract:** We present two architectures for implementing optical buffers. Both use QD-SOA devices as wavelength converters and fixed-length delay lines that are combined to form both output queued and "parallel buffer" switches. Scheduling algorithms are proposed to prevent packet collision and the computer simulation results are also given.
**Keywords:** Optical packet switching, optical buffering, packet scheduling, delay lines.

## 1. Introduction

In this paper we present novel optical switch architectures. There are many important issues that we have to concern to implement this architectures in a real-world system. As we all know, an optical transmission is currently a common standard. To obtain best results we have also to try omitting an optical-electronic-optical conversion in switching devices. Though many progresses are done now, all-optical packet (burst) switching systems are only in development phase. We present two architectures for implementing optical buffers. Both use QD-SOA devices as wavelength converters and fixed-length delay lines that are combined to form both output queued and "parallel buffer" switch. QD-SOA (Fig. 1) acts like a wavelength converter but as opposed to earlier presented SOA`s it can operate on WDM signals. It can be used both in a buffering stage and a switching stage. More detailed description of this device is provided in [1]. Physical layer simulation of an optical buffer based on QD-SOA`s is carried out in [4].

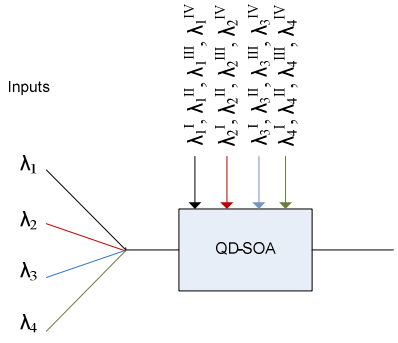



Fig. 1. QD-SOA model

## 2. Switching architectures

The first of presented architectures is based on output queuing and depicted in Fig. 2. Based on [1], we assumed that WDM input signal consists of 4 wavelengths: $\lambda_1 \ldots \lambda_4$. First QD-SOA changes wavelength so that packets enter the branch that correspond to the proper output.

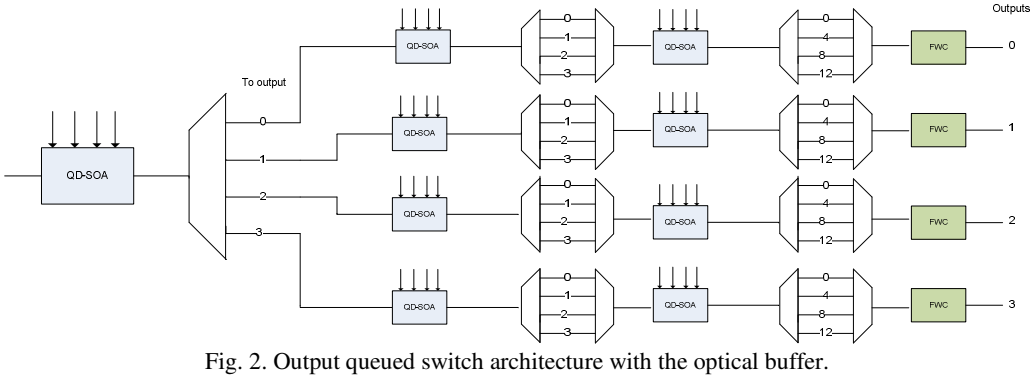



Fig. 2. Output queued switch architecture with the optical buffer.

Switching control should be easy to implement and done in a very fast way. To achieve maximum simplicity, we use counters as the main component in the scheduling process. We assume that every output has its own counter that shows the current buffer state (a queue length). An example of the counter is shown in Table 1. The scheduling algorithm is described below.

| Output number | *1* | *2* | *3* | *4* |
|---|---|---|---|---|
| Delayed packets [counters] | **3** | **2** | **2** | **0** |

Tab.1. An array representation of counters.

There are *n* counters; each of them shows how many packets are in the queue to the given output. In the beginning we started from resetting all counters. In every time slot we decrease all counters by one (if counter > 0) and check if any packet appears at any input. If a new packet appears we check the counter, determine a proper delay, and increase the counter by one. In every time slot we may begin to search from the input next to the last searched, so that every input will have the same priority. At the end of every time slot, we decrease all counters by one.

Counter-based scheduling algorithm:

1. In the first time slot set all counters on zero
2. Set *i*=0,
3. Check if there is a new packet in *i*-th input
4. If there is a new packet,
    a. Determine its delay by counter value.
    b. Change counter value:
    new counter value = old counter value + packet length
5. If not, i = i+1 and go to step 3. If all inputs are checked, go to step 6
6. Set QD-SOA's,
7. Decrease all counters by one
8. Go to next time slot and start from step 2.

The presented architecture has many advantages. First of all, it is an output buffering switch which is characterized by the best queuing properties. Using 4 internal wavelengths for one input wavelength, we can obtain the 15 time slots buffer in 2 QD-SOA sections. Adding one more QD-SOA extends buffer to 63 time slots (Fig. 3). The proper buffer size depends on traffic conditions as well as the size of the switch.

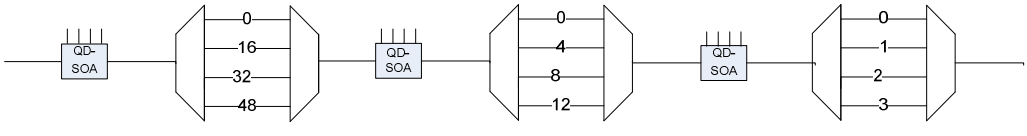



Fig. 3. Proposed optical buffer in output queued switch

One of the interesting functions of QD-SOA is that it may copy one signal to more than one wavelength, so packets can be distributed over many outputs. As such we are able to apply the multicast feature in this architecture. There is also no need to segment packets into smaller fixed-size cells.

Other interesting architecture is depicted in Fig 4. Buffer size is equal to 15 time slots. We assume that input WDM signal consists of 4 wavelengths: $\lambda_1 \ldots \lambda_4$ and is introduced to QD-SOA. Every control signal convert input wavelength to one of four wavelengths so that the optical signal goes through the relevant MUXs output and reaches the next QD-SOA. The second QD-SOA converts the signal to achieve desirable delay. After the buffering stage, a switching should be performed. It can be accomplished by combining an AWG and a QD-SOA.

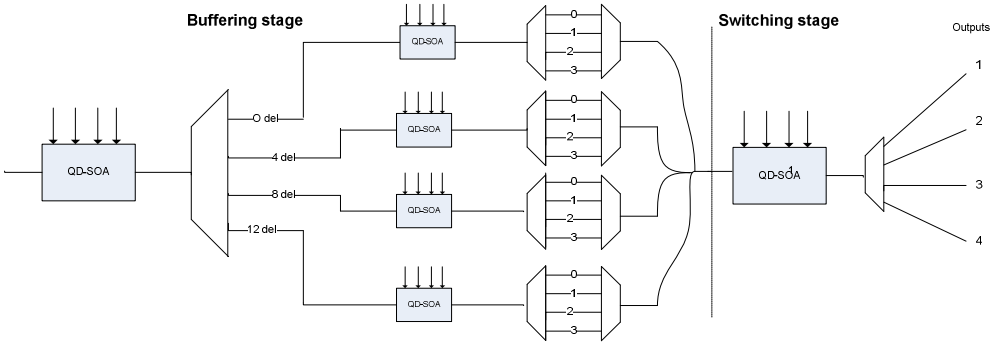



Fig. 4. Switch architecture with parallel, optical buffer.

The main problem arises when 2 or more packets from the same input go to different outputs but arrive to the same QD-SOA simultaneously. Counter-based scheduling algorithm doesn`t take it into consideration and the packet contention appears. In the proposed "parallel buffer" architecture, there will be no contention in the buffering stage but could appear in the switching stage.

Let us consider the example shown in Figures 5-7. In x-th time slot only packet $P_{1,2}$ (from the first input to the second output) enter the switch. It goes through the third branch and then enters the first delay line. Six time slots later, packet $P_{1,4}$ enters the switch. The route for this packet within the buffering stage is shown in Fig.6. In (x+8)-th. time slot both packets enter the switching stage. QD-SOA will not serve two signals from the same input at the same time. Contention will occur because this is not possible to convert e.g. $\lambda_1^{I}$ and $\lambda_1^{II}$ simultaneously in one QD-SOA device. Of course, it is possible that more than 2 packets will compete in the QD-SOA. This situation is analogous to that considered earlier and will not be further discussed.

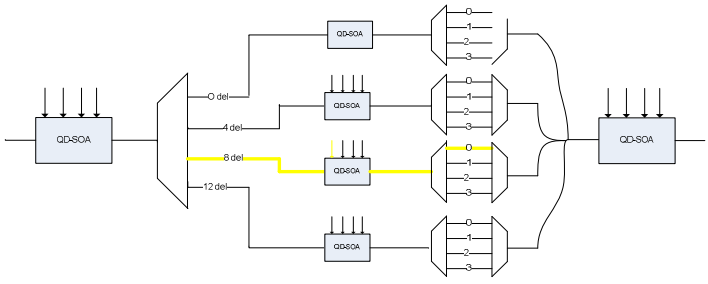



Fig. 5. Packet $P_{1,2}$ enter the switch in x-th. time slot.

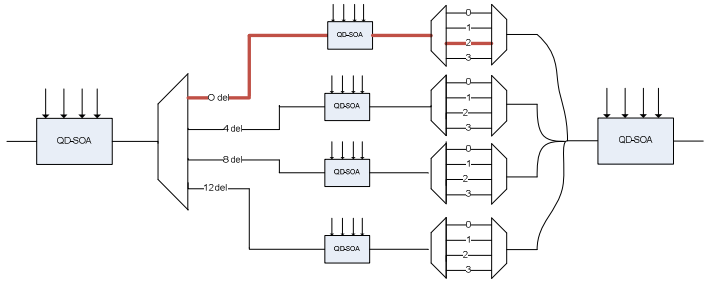



Fig. 6. Packet $P_{1,4}$ enter the switch in (x+6)-th. time slot.

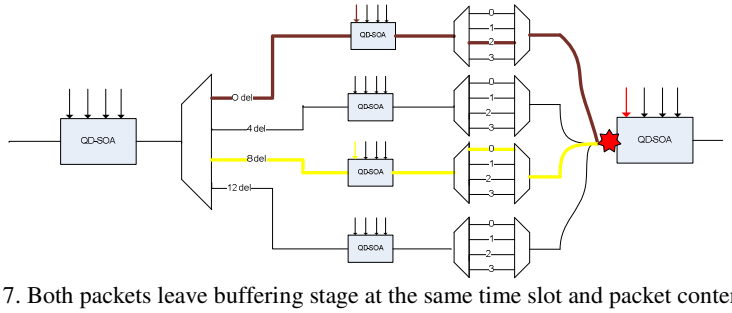



Fig. 7. Both packets leave buffering stage at the same time slot and packet contention appears on the input of the third QD-SOA

To overcome this problem we propose the following solutions. One of them is to use more internal wavelengths (hardware dependent); the second one is more sophisticated and will be described below.

We assume that packets are of fixed size and their duration is one time slot. Every input has its own array, where the proper delay is computed. When the packet enters the switch and its delay is set, it is marked by “X” in a relevant time slot of the array. At the same time we should prevent packet contention from the same input, so we mark “F” in all other cells corresponding to this time slot. Next, we are searching for the next free

time slot, which we can sent packet to this output and mark it by “P”. Then we mark that this time slot to given output is busy now and shouldn`t be used by other inputs Appropriate pointers are set on the next empty field. At the beginning of the next time slot, all values are moved one step left. Fifteen time slots array for the first and the second input in some x-th. time slot is shown in Tab. 2 and Tab. 3, respectively.

| Output/TS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **B** | F | F | **X** | **B** | **B** | **P** | | | | | | | | |
| 2 | **P** | F | F | F | | | | | | | | | | | |
| 3 | | F | **X** | F | **P** | | | | | | | | | | |
| 4 | | **X** | F | F | **P** | | | | | | | | | | |

Tab. 2. Time slot occupancy for the first input

| Output/TS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **X** | **B** | F | **B** | **X** | **B** | **P** | | | | | | | | |
| 2 | F | **P** | F | F | F | | | | | | | | | | |
| 3 | **F** | **P** | **B** | F | F | | | | | | | | | | |
| 4 | F | **B** | **X** | **X** | F | **P** | | | | | | | | | |

Tab. 3. Time slot occupancy for the second input

X – in this time slot, a packet leaves the buffer
F – this time slot is forbidden for this input because another packet from this input leaves the switch.
B – indicates that some other input uses this time slot
P – points to the first time slot in which we can sent packet to relevant output

The parallel buffer architecture has many advantages. First of all, we can implement 15 time slots buffer using limited number of QD-SOA sections. This is very important in optical domain systems because of optical signal to noise ratio. Total number of QD-SOA is 5 but signal has to go only through 2 QD-SOA stages. We have assumed that an input signal consist on $w = 4$ different wavelengths. It is worth noticing that the number of input wavelengths depends on QD-SOAs and will be higher as technology will be more mature. The total size of the buffer depends on $w$ and is equal to $w^2$-1. Presented buffer architecture could be more cascaded. An example of the buffer of size 31 is depicted in Fig. 8.

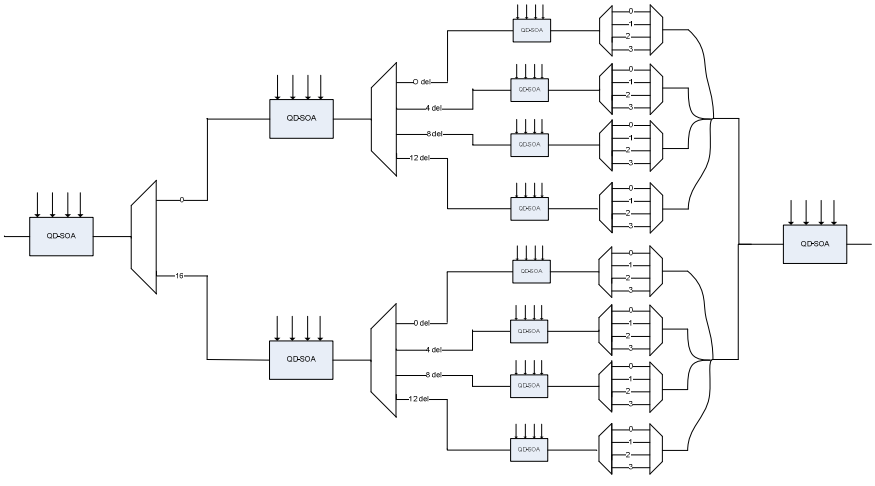



Fig. 8. Switch architecture with parallel, optical buffer, *b*=31

## 3. Simulation experiments

In simulation experiments, we consider the Bernoulli arrival model where cells arrive at each input in slot-by-slot manner. Under the Bernoulli arrival process, the probability that there is a cell arriving in each time slot is identical and independent of any other slot. The probability that a cell may arrive in a time slot is referred to as the load of the input. The experiments have been carried out for a wide range of traffic load: from 0.05 to 0.9 with the step 0.05. Time preceded the exact simulation was 10000 cycles and exact simulation lasts 90000 time slots. The results are presented in charts 9-12 and ranged from 0.5 to 0.9. The lower load was omitted because of small measured values. We have evaluated following performance measures: a maximum occupancy of the buffer and a mean cell delay in case of b = infinity as well as the cell loss probability and mean cell delay when b = 15, 31 and 63. "Parallel buffer" architecture suffers from a higher mean cell delay and maximum buffer occupancy. That is obvious, because output queued switches are referred to as ideal. It is worth noticing that in case of "parallel buffer" architecture for the load lower than or equal to 0.7 (b = infinite) the mean cell delay was less then 3 time slots, which may be considered as very satisfactory. For the load less then 0.85 (b = infinite) mean delay do not exceed 8 time slots, regardless of the switch. Maximum buffer occupancy for N=16 in case of output buffered switch was 45 cells and for the "parallel buffered" switch was 63. Thus there was no losses in both architectures and measures was analogous to the simulation with b = infinite. Applying 16. time slots buffer, losses have shown up under the load equal to 0.6. Increasing the size of the buffer to 31, packet losses appears under the load equal 0.8. Simulation results for the architectures when b=31 are shown in Tab 4. It is worth noticing that considering QD-SOA`s devices which convert each signal to one of four wavelengths and using buffer in

the first proposed switch architecture, we cannot implement 31 time slot buffer. In that case, simulation was prepared only for theoretical comparative purposes.

| Architecture | Size | Load | CLP | Average delay |
|---|---|---|---|---|
| Parallel | 4x4 | 0,9 | 2,40E-04 | 9,96 |
| Output | 4x4 | 0,9 | 2,49E-05 | 3,36 |
| Parallel | 8x8 | 0,85 | 3,23E-05 | 7,6339 |
| Output | 8x8 | 0,85 | 1,47E-06 | 2,48 |
| Parallel | 8x8 | 0,9 | 9,90E-04 | 12,41 |
| Output | 8x8 | 0,9 | 9,03E-05 | 3,99 |
| Parallel | 16x16 | 0,8 | 4,69E-06 | 5,61 |
| Parallel | 16x16 | 0,85 | 7,57E-05 | 8,36 |
| Output | 16x16 | 0,85 | 2,20E-06 | 2,65 |
| Parallel | 16x16 | 0,9 | 1,50E-03 | 13,58 |
| Output | 16x16 | 0,9 | 1,10E-04 | 4,19 |

Tab. 4. Cell loss probability and average cell delay, $b$ = 15

## 4. Conclusion

In this paper we have proposed novel optical switch architectures using QD-SOA devices. Both architectures achieve very interesting results, considering both mean cell delay and maximum buffer occupancy. The 4x4 output buffered switch in case of large buffers can be implemented using fewer QD-SOA devices than "parallel buffered" switch. Nevertheless, the number of necessary hardware is dependent on the number of outputs. When QD-SOA devices become more mature and allow to simultaneous convert more than 4 input signals, "parallel buffer" architecture become more profitable, because the number of used QD-SOA depends only on the buffer size. At the moment we are working on the multicast feature and adapting the algorithm used in "parallel buffer" architecture to optical burst switching.

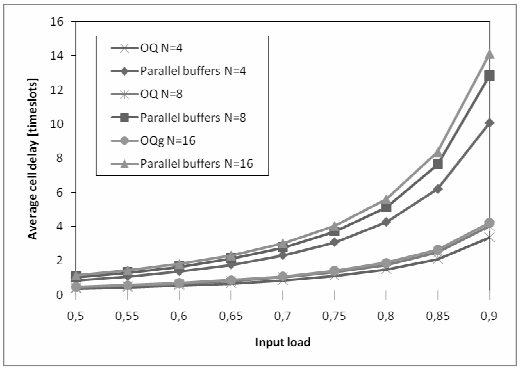



Fig. 9. Average cell delay.$b$ = infinite

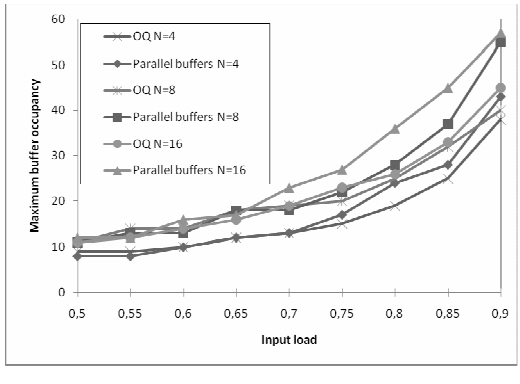



Fig. 10. Maximum buffer occupancy. $b$ = infinite

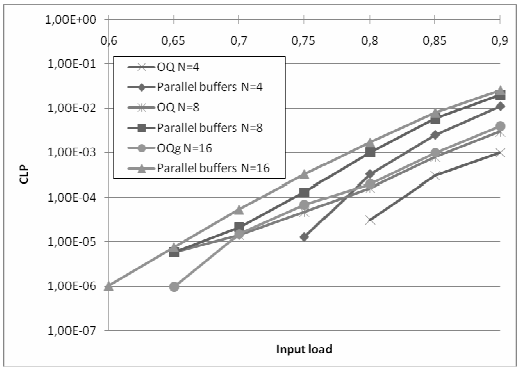



Fig. 11. Cell loss probability, b=15

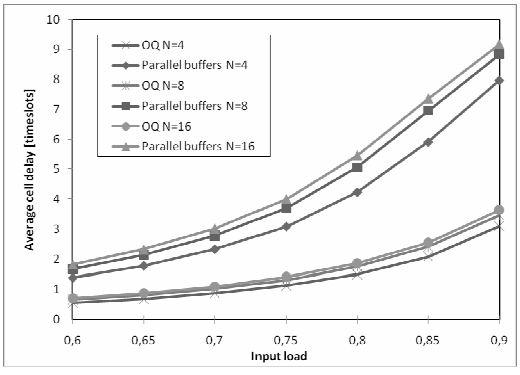



Fig. 12. Average cell delay, $b$ = 15

# References

[1] Spyropoulou M., Sygletos S., Tomkos I.,; *Investigation of Multi-Wavelength Regeneration Employing Quantum-Dot Semiconductor Optical Amplifiers beyond 40Gb/s.* International Conference on Transparent Optical Networks, 2007, Volume 1, Pages:102 – 105, July 2007

[2] Danilewicz G., Głąbowski M., Kabaciński W., Kleban J.: *Packet switch architecture with Multiple Output Queueing*, Globecom, . Volume 2, Pages:1192 – 1196 Nov.-3 Dec. 2004.

[3] Baranowska A., Danilewicz G., Kabaciński W., Kleban J., Parnewicz D., Dąbrowski P.: *Performance Evaluation of the Multiple Output Queueing Switch Under Different Traffic Patterns*. Globecom, Volume 1, 28 Nov.-2 Dec. 2005.

[4] Spyropolou M., Yiannopolous K., Sygletos S.,Vlachos K., Tomkos I.: *A160 Gbps Simulation of a Quantum Dot Semiconductor Optical Amplifier Based Optical Buffer*. Optical Network Design and Modelling,Athens, May 2007.

[5] Konstantinos I., Yiannopolous K, Vlachos K., Varvarigos E.: *TheMultiple Input Buffer and Shared Buffer Architectures for Asynchronous Optical Burst Switching Network*. In Journal of Lightwave Technology, vol 25, Issue 6, Pages: 1379 – 1389, June 2007.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 53 - 64

# Wide-sense Non-blocking Multirate and Multicast $\log_2(N; m; p)$ Switching Networks

WOJCIECH KABACIŃSKI[1] TOMASZ WICHARY [2]

[1] Faculty of Electronics and Telecommunications
Poznan University of Technology
*wojciech.kabacinski@et.put.poznan.pl*

[2] Nokia Siemens Networks Polska Sp. z o.o.
ul. Sienna 73, 00-833 Warszawa
*tomasz.wichary@nsn.com*

**Abstract:** In this paper we are going to investigate the non-blocking operation of multirate $\log_2(N; m; p)$ switching networks with multicast connections. The nonblocking operation of multi-$\log_2 N$ switching networks for point-to-point connections was firstly considered in [5], [6]. Conditions under which these switching networks are nonblocking for multicast connections were proposed by Tscha and Lea [7] and corrected by the same authors in [8]. They considered a wide-sense nonblocking switching networks and proposed a control algorithm based on the blocking window of a given size. Their results were later generalized by Danilewicz and Kabaciński to blocking windows of any size [9], [10], [11]. Multirate multi-$\log_2 N$ switching networks for point-to-point connections were considered by Lea in [14] and by Kabaciński and Żal in [15]. Frank K. Hwang and Bey-Chi Lin in [17] made an attempt to extended these connections to multicast $\log_2(N; m; p)$ networks but encountered some difficulties which were commented by Danilewicz and Kabaciński in [18] where the correct results were given. In this paper we will extend these known results to multirate c switching networks with multicast connections and this is the continuation of results given in [16], [17] and [18]. The nonblocking operation at the connection level is considered.

**Keywords:** wide-sense nonblocking networks, multicast, multirate, multi-$\log_2 N$ switching networks

## 1. Introduction

In future telecommunication networks large capacity routers will be needed. These routers will use high-capacity and high-speed switching networks based on optical transmission units and optical switching elements. Many papers have been published concerning the space and time domain as well as the wavelength domain (WDM) switching [1], [2], [3], [19], [20].

One of the architecture considered for both high-speed electronic and photonic switching is a switching network composed of $\log_2(N; m; p)$ stages. However, the main drawback of such architecture is its blocking characteristics. To create nonblocking architecture two methods were proposed: vertical stacking and horizontal cascading. Switching networks obtained by vertically stacking $p$ copies of $\log_2(N; m; p)$ switching

networks are called multi-$\log_2 N$ switching networks. Nonblocking operation of such networks for point-to-point connections were considered in [5], [6]. Horizontal cascading method introduces a greater number of stages between each inlet-outlet pair. These additional stages introduce a greater delay. The major benefit of this method is the lower cost than in the vertical stacking method. The switching network obtained by the vertical stacking method is called the multi-$\log_2$N switching network. It consists of $p$ copies of vertically stacked $\log_2(N; m; p)$ networks. The benefit of VS method is the same number of stages. Conditions under which these switching networks are nonblocking for multicast connections were proposed by Tscha and Lea [7] and corrected by the same authors in [8]. They considered a wide-sense nonblocking switching networks and proposed the control algorithm based on the blocking window of a given size. Their results were later generalized by Danilewicz and Kabaciński to the blocking window of any size [9], [10], [11]. Multirate multi-$\log_2 N$ switching networks for point-to-point connections were considered by Lea in [14] and by Kabaciński and Żal in [15].

In [16] were given results for $\log_2(N; 0; p)$ switching networks and discrete bandwidth case, when $1 \le t \le \lfloor n/2 \rfloor$. In [17] were given results for $\log_2(N; 0; p)$ switching networks and continuous bandwidth case, for $B \in (1-b;\beta]$ and $B \in (1-2b;1/2]$, $b \in (1/4;1/2]$ sub cases. In [18] were given results for $\log_2(N; 0; p)$ switching networks and discrete bandwidth case , when $\lfloor n/2 \rfloor + 1 \le t \le n-1$. In this paper nonblocking conditions for multirate multicast connections will be considered and the results will be extended for $\log_2(N; m; p)$ switching networks and discrete bandwidth case, when $1 \le t \le \lfloor n/2 \rfloor$.

The paper is organised as follows. In Section 2, the model used in this paper is described. In Section 3, non-blocking conditions for $\log_2(N; m; p)$, $1 \le t \le \lfloor n/2 \rfloor$ for discrete bandwidth case are given and proved. In Section 4 example of the worst state of $\log_2(32;1;16)$ switching networks is shown. In Section 5 numerical results and analysis were given. Finally, concluding remarks were given.

## 2. Model description

Multi-$\log_2 N$ switching networks are constructed by vertically stacking $p$ copies of $\log_2(N; m; p)$ networks. Each of $\log_2(N; m; p)$ switching networks is called a plane and is composed of 2 x 2 switches arranged in $\log_2(N; m; p)$ stages. There are some equivalent topologies [6]. The topology of one network can be transform to another network by changing the position of the switches. These networks are self-routing.

Similarly as in [3], [4], and [7] we will use the algorithm based on blocking windows. In this concept a set of all outputs is divided into subsets. Each such subset is called a blocking window (*BW*). A new multicast connection is divided into subconnections, were outputs requested in each subconnection belonging to one blocking window.

*Definition 1.* Let the set of inputs $O = \{ 0, 1, 2, ..., N-1 \}$ will be divided on $N/K$ subsets $O_i = \{ K \cdot i, K \cdot i + 1, K \cdot i + 2, ..., K \cdot i + (K-1) \}$, where $i \in [0; N/K - 1]$, $K = 2^t$, $t \in [1; n]$. Each of subsets $O_i$ will be called the blocking window (BW).

Throughout the paper we will use following notations: $NBW_i$ (number of blocking windows), $NCP_i$ (number of connection paths), $SI_i$ (accessible inputs), $SO_i$ (accessible outputs), $MMC$ (maximum multicast connection). Definitions of these elements can be found in [16], [17]. We will use also the bipartite graph representation described in these papers.

Let's $\langle x; Y; \omega \rangle$ be a new connection. This connection will go through vertices *x* and *y* in stages 0 and *n*, respectively, and some vertices in stages from 1 to $n-1$. When circuit switching is used, no paths are allowed to intersect at any vertex in one plane. In the case of multirate switching more connections may be realized through the same vertex provided that the sum of weights of these connections will be less than or equal to 1 (i.e. to the corresponding inter-stage link capacity). In stage 1, connection $\langle x; Y; \omega \rangle$ may be blocked by connections $\langle k; l; \eta \rangle$ (where $k \in SI_1$, $l \in SO_1$) from one new input, when the total bandwidth of these connections is greater than $1-\omega$. In stage 2, this connection may be blocked by connections $\langle k; l; \eta \rangle$ (where $k \in SI_2$, $l \in SO_2$) from two new inputs, when the total bandwidth of these connections is greater than $1-\omega$. Generally, a new connection may be blocked in stage *i* (where $1 \le i \le n-1$) by connections $\langle k; l; \eta \rangle$ (where $k \in SI_i$, $|SI_i| = 2^{i-1}$, $l \in SO_i$, $|SO_i| = 2^{n-i-1}$), when the total bandwidth of these connections is greater than $1-\omega$.

We will use the following algorithm for setting up a new connection $\langle x; Y; \omega \rangle$.

Algorithm 1:

*Step 1.* Divide connection $\langle x; Y; \omega \rangle$ to subconnections $\langle x; Y_i; \omega \rangle$ such that $Y_i$ contains all possible outputs of set *Y* belonging to the same blocking window $O_i$.

*Step 2.* Try to set up the subconnection though one of already occupied planes starting from plane used for setting up the last connection.

*Step 3.* If Step 2 failed set up the subconnection through one of free planes (not occupied planes).

## 3. Non-blocking conditions

We will prove the sufficient nonblocking condition by showing the worst state of the switching network. Let $\langle x; Y; \omega \rangle$ is a new connection. This connection may be point-to-

point connection or a part of a multicast connection. This connection may be blocked by $\left(\lfloor(1-\omega)/b\rfloor+1\right)=\lfloor(1-\omega+b)/b\rfloor$ connections of weight $b$, set up in the same node.

In each of $N$ inputs in stage 0 and $N$ outputs in stage $n$ up to $\lfloor\beta/b\rfloor$ connections of weight $b$ can be set up. In input $x$ and in output $y$ up to $\lfloor(\beta-\omega)/b\rfloor$ connections of weight $b$ can be set up. In the switching network a connection of weight $b_k$, $b_k=1,2,\ldots K,$ is an equivalent to $b_k/b$ connections of weight $b$. On the assumption that we use only connections of weight $b$, we do not loose the generality of the approach.

In the following Theorem we will use following equations:

$$\alpha(\omega)=\lfloor(1-\omega+b)/b\rfloor,\quad R\left(SI_0^m;\omega\right)=\lfloor(\beta-\omega)/b\rfloor, \tag{1, 2}$$

$$R\left(SI_i^m;\omega\right)=\left(\left|SI_i^m\right|\lfloor\beta/b\rfloor+R\left(SI_{i-1}^m\right)\right)\otimes\alpha(\omega),\quad R\left(SO_{n+m}^m;\omega\right)=\lfloor(\beta-\omega)/b\rfloor, \tag{3, 4}$$

$$R\left(SO_{n+m-i}^m;\omega\right)=\left(\left|SO_{n+m-i}^m\right|\lfloor\beta/b\rfloor+R\left(SO_{n+m-i+1}^m\right)\right)\otimes\alpha(\omega), \tag{5}$$

$$Q_{n+m-1}(\omega)=R\left(SO_{n+m-2}^m\right), \tag{6}$$

$$Q_{n+m-i}(\omega)=\left(2\cdot Q_{n+m-i+1}(\omega)\right)\otimes\alpha(\omega),\text{where } 2\le i\le t, \tag{7}$$

$$\gamma_{SI}(i;\omega)=\left\lfloor\frac{\left|\boldsymbol{SI}_i^m\right|\lfloor\beta/b\rfloor+R\left(\boldsymbol{SI}_{i-1}^m;\omega\right)}{\alpha(\omega)}\right\rfloor=\left\lfloor\frac{2^{i-1}\lfloor\beta/b\rfloor+R\left(\boldsymbol{SI}_{i-1}^m;\omega\right)}{\alpha(\omega)}\right\rfloor, \tag{8}$$

$$\gamma_{SO}(i;\omega)=\left\lfloor\frac{\left|\boldsymbol{SO}_{n+m-i}^m\right|\lfloor\beta/b\rfloor+R\left(\boldsymbol{SO}_{n+m-i+1}^m;\omega\right)}{\alpha(\omega)}\right\rfloor=\left\lfloor\frac{2^{i-1}\cdot\lfloor\beta/b\rfloor+R\left(\boldsymbol{SO}_{n+m-i+1}^m;\omega\right)}{\alpha(\omega)}\right\rfloor, \tag{9}$$

where $a\otimes b$ means the rest of dividing $a$ by $b$.

**Theorem 1.** *The $\log_2(N; m; p)$ multirate switching network is nonblocking in the wide sense for discrete bandwidth case when algorithm 1 is used if and only if:*

A: for $m=1$, $1\le t<\lfloor n/2\rfloor$; $m=1$, $t=\lfloor n/2\rfloor$, $n$ is odd; $m=2$, $2\le t\le\lfloor n/2\rfloor$ or $2<m\le t$, $t\le\lfloor n/2\rfloor$

$$p\ge\left\lfloor\sum_{i=1}^{m}\gamma_{SI}(i;B)\cdot2^{n-t-i}+v\cdot\left(\sum_{i=m+1}^{t}\gamma_{SO}(i;B)\cdot2^{-m}+\sum_{i=m+1}^{t}\gamma_{SI}(i;B)\cdot\left(2^{n-t-i}-2^{-m}\right)\right)+\right.$$

$$\left.+\left\lfloor\left(\left[R\left(SO_t^m;B\right)+2^t\lfloor\beta/b\rfloor-\left\lfloor\frac{2^t\lfloor\beta/b\rfloor-B/b}{\alpha(B)}\right\rfloor\alpha(B)\right]\right)\Big/\alpha(\omega)\right\rfloor\cdot2^{-m}\right\rfloor+1, \tag{10}$$

where $v=0$ for $t\le m$, $v=1$ for $t>m$,

B: for $m=1$, $t=n/2$, $n$ is even

$$p \ge \left\lfloor \gamma_{SI}(1;B)\cdot 2^{n-t-1} + v\cdot\sum_{i=2}^{t}\gamma_{SI}(i;B)\cdot 2^{n-t-i} \right\rfloor + 1, \tag{11}$$

C$^{(*1,\,*2)}$: for $m\ge 2$, $t=1$; $m>2$, $t=2$ $^{(*1)}$ or $m>2$, $2<t<m$, $t\le\lfloor n/2\rfloor$ $^{(*2)}$

where (*1) and (*2) denote that the part of equation (12); and when is used for cases denoted by (*1) and (*2), respectively

$$\begin{aligned} p \ge \Bigg\lfloor & \left(\sum_{i=1}^{t}\gamma_{SI}(i;B)\cdot 2^{n-t-i}\right)^{(*1)} + \left\lfloor \frac{\lfloor\beta/b\rfloor + (2^t-1)\cdot\lfloor\beta/b\rfloor\otimes\alpha(B)}{\alpha(B)} \right\rfloor \left(2^{n-2t-1}-2^{-(t+1)}\right) + \\ & + \left(2^{t-2}\left\lfloor \frac{\lfloor\beta/b\rfloor + R\left(SO_{n+m}^{m};B\right)}{\alpha(B)} \right\rfloor + \sum_{i=2}^{t} 2^{t-2i}\left\lfloor \frac{2\cdot Q_{n+m-i+1}(B)}{\alpha(B)} \right\rfloor\right)^{(*2)} + \\ & + \sum_{i=1}^{m-t}\left\lfloor \frac{V^{i-1}(B) + 2^{i-1}\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(B)\right)}{\alpha(B)} \right\rfloor \cdot 2^{-(t+i)} + \\ & + \sum_{i=m-t+1}^{n-t-1}\left\lfloor \frac{V^{i-1}(B) + 2^{i-1}\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(B)\right)}{\alpha(B)} \right\rfloor \cdot 2^{-m} + \\ & + \sum_{i=n-t}^{n+m-2t+1}\left\lfloor \frac{V^{i-1}(B) + 2^{i-1}\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(B)\right)}{\alpha(B)} \right\rfloor \cdot 2^{-(n+m-t-i)} \Bigg\rfloor, \end{aligned} \tag{12}$$

where $V^0(\omega) = R\left(SO_t^m;\omega\right)^{(*1)}$, $V^0(\omega) = Q_{n+m-t}(\omega)^{(*2)}$,

$$V^i(\omega) = \left(V^{i-1}(\omega) + 2^{i-1}\cdot\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right)\right)\otimes\alpha(\omega),$$

$$Q_{n+m-1}(\omega) = R\left(SO_{n+m-1}^{m};\omega\right),\ Q_{n+m-i}(\omega) = 2\cdot Q_{n+m-i+1}(\omega)\otimes\alpha(\omega),$$

D: for $m>2$, $2<m\le t\le\lfloor n/2\rfloor$

$$\begin{aligned} p \ge \Bigg\lfloor & 2^{t-2}\left\lfloor \frac{\lfloor\beta/b\rfloor + R\left(SO_{n+m}^{m};B\right)}{\alpha(B)} \right\rfloor + \sum_{i=2}^{t} 2^{t-2i}\left\lfloor \frac{2\cdot Q_{n+m-i+1}(B)}{\alpha(B)} \right\rfloor + \sum_{i=1}^{m}\frac{\gamma_{SI}(i;B)}{2^i}\left(2^{n-t}-1\right) + \\ & + \sum_{i=m+1}^{t}\gamma_{SI}(i;B)\left(2^{n-t-i}-2^{-m}\right) + \left\lfloor \frac{\lfloor\beta/b\rfloor + (2^t-1)\lfloor\beta/b\rfloor\otimes\alpha(B)}{\alpha(B)} \right\rfloor \left(2^{n-2t-1}-2^{-m}\right) + \\ & + \sum_{i=1}^{n+m-2t+j}\left\lfloor \frac{V^{i-1}(B) + 2^{i-1}\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(B)\right)}{\alpha(B)} \right\rfloor \cdot 2^{-m} \Bigg\rfloor. \end{aligned} \tag{13}$$

***Proof***. Let $\langle x;y;\omega\rangle$ be a new connection. This connection may be a point-to-point connection or to be part of multicast connection. This connection can be blocked by connections of weight greater then $1-\omega$ which passes through any of nodes belonging to the path of the new connection.

This connection may be blocked by connections $\langle k;l;b\rangle$ in stage $i$, $1\le i\le t$, $k\in SI_{n-i}$, $l\in\{y\}\cup\bigcup_{j=1}^{i}SO_{n-j}^{m}$. Since one path of this new connection in one plane is blocked by $\alpha(\omega)$ such connections $\langle k;l;b\rangle$ and we have $NCP_{n+m-i}$ in nodes of stage $n+m-i$, therefore the number of blocked planes is given by $p_1$, where

$$p_1=\sum_{i=1}^{t}\frac{\gamma_{SO}(i;\omega)}{NCP_{n+m-i}} \tag{14}$$

It should be noted that for cases C$^{(*2)}$ and D we have $m\ge 2$; a new connection is set up to $|Y|=2^{t-2}$, $Y\in BW_0$ outputs. In these cases the new connection in a multicast connection $\langle x;Y;\omega\rangle$, $|Y|=2^{t-2}$, similarly as it was described in [13]. The highest number of plane is blocked in the nodes of stage $n+m-1$. However we should also consider some remind bandwidth on outputs, which will not block the plane in node of stage $n+m-1$, but the sum of these remain bandwidth my be cumulated in the nodes of stages $n+m-2$ and earlier. In these stages $n+m-i$, $2\le i\le t$ we may have $Q_{n+m-i}(\omega)=2\cdot Q_{n+m-i+1}(\omega)\otimes\alpha(\omega)$ connections which may block $2^{t-i}\lfloor 2\cdot Q_{n+m-i+1}(\omega)/\quad/\alpha(\omega)\rfloor/NBW_{n+m-i}$ planes, so the total number of planes $p_1$ is given by:

$$p_1=2^{t-1}\frac{\left\lfloor\dfrac{\lfloor\beta/b\rfloor+R(SO_{n+m}^{m};\omega)}{\alpha(\omega)}\right\rfloor}{NCP_{n+m-1}}+\sum_{i=2}^{t}2^{t-i}\frac{\left\lfloor\dfrac{2\cdot Q_{n+m-i+1}(\omega)}{\alpha(\omega)}\right\rfloor}{NCP_{n+m-i}}. \tag{15}$$

At stages from 1 to $t$ we will consider multicast connections. At the stage $i$, $1\le i\le t$ connections $\langle k;l;b\rangle$, $k\in\{x\}\cup\bigcup_{j=1}^{i}SI_j^m$ may be set up to $|L|\in NBW_i$ blocking windows. The rest of free bandwidth may be used by connections at stage $i+1$. Generally, at stage $i$, connections $\langle k;l;b\rangle$ may block $\gamma_{SI}(i;\omega)\cdot(2^{n-t-i}-1)$ planes for a new one. In the worst state they may block $p_2$ planes, where

$$p_2=\sum_{i=1}^{t}\frac{\gamma_{SI}(i;\omega)}{NCP_i}(NBW_i-1), \tag{16}$$

At stage c we may set up multicast connections to accessible blocking windows. In case A we may consider connections to $BW_0$ and $BW_1$. To $BW_0$ we may still set up $R(SO_t^m;\omega)$ connections of weight $b$. To $BW_1$ we may still set up $\lfloor(2^t\lfloor\beta/b\rfloor-\omega/b)/\alpha(\omega)\rfloor\cdot\alpha(\omega)$

connections of weight $b$. Connections set up to this free bandwidth may block next planes for the new one. In the worst case these connections may block $p_3$ planes, where

$$p_3 = \frac{R\left(SO_t^m;\omega\right)+2^t\lfloor\beta/b\rfloor-\left\lfloor\dfrac{2^t\lfloor\beta/b\rfloor-\omega/b}{\alpha(\omega)}\right\rfloor\alpha(\omega)}{NCP_{t+1}} \tag{17}$$

In cases C and D we may realize multicast connections to $BW_i$ through the node of stage $t+1$, where $i\in\langle 1;2^{n-t}-1\rangle$, $i\in N$. There is at least free one totally free output and free bandwidth for $\left(2^t-1\right)\lfloor\beta/b\rfloor\otimes\alpha(\omega)$ connections of weight $b$ in the other outputs in the same blocking window. Theses connections together may block $p_3$ planes, where

$$p_3 = \frac{\left\lfloor\dfrac{\lfloor\beta/b\rfloor+\left(2^t-1\right)\lfloor\beta/b\rfloor\otimes\alpha(\omega)}{\alpha(\omega)}\right\rfloor}{NCP_{t+1}}\left(NBW_{t+1}-1\right) \tag{18}$$

It should be noted that mentioned-above connections do not exist for case B.

In the blocking windows we may still have a free bandwidth. In $BW_0$ we may still set up $R\left(SO_t^m;\omega\right)$ connections of weight $b$. In the rest of $BW_i$, where $i\in\langle 1;2^{n-t}-1\rangle$, $i\in N$ we may set up $2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)$ connections of weight $b$. Connections set up to through stage $n+m-t$ to outputs belonging to $BW_0$ do not block any additional plane. Connections set up through stage $n+m-t-1$ to the free bandwidth in outputs belonging to $BW_0$ and $BW_1$ may block $\left\{V^0(\omega)+2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right\}/\alpha(\omega)$ planes for the new connection. In these outputs we may still have a free bandwidth and we may use it for $V^1(\omega)=\left(V^0(\omega)+2^0\left\{2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right\}\right)\otimes\alpha(\omega)$ connections of weight $b$. Connections set up through stage $n+m-t-2$ may block $\left(V^1(\omega)+2^1\left\{2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right\}\right)/\alpha(\omega)$ planes for the new connection. In the outputs belonging to accessible blocking windows may bandwidth still be free and we may have $V^2(\omega)=\left(V^1(\omega)+2^1\left\{2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right\}\right)\otimes\alpha(\omega)$ connections of weight $b$. Generally, these connections may be realized through stages from $n+m-t-1$ to $t+1$ to outputs belonging to $BW$. They may block $\left(V^{i-1}(\omega)+2^{i-1}\cdot\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right)\right)/\alpha(\omega)$, $1\le i\le n+m-2t-2$ planes for a new one (where for stage $n+m-t-1$ we have $V^1$, for $n+m-t-2$ we have $V^2$, …, for $t+1$ - $V^{n+m-2t-1}$). On outputs belonging to accessible $BW$ from stage $n+m-t-i$, we may still have free bandwidth $V^i(\omega)=\left(V^{i-1}(\omega)+2^{i-1}\cdot\left(2^t\lfloor\beta/b\rfloor\otimes\alpha(\omega)\right)\right)\otimes\alpha(\omega)$. To this outputs we may realize connections through previous stages.

In the worst state, connections to this free bandwidth at outputs belonging to accessible $BW_i$, $t+1\le i\le n+m-t$ may block $p_4$, where

$$p_4 = \sum_{i=1}^{n+m-2t-1} \frac{\left\lfloor \dfrac{V^{i-1}(\omega) + 2^{i-1}\left(2^t \lfloor \beta/b \rfloor \otimes \alpha(\omega)\right)}{\alpha(\omega)} \right\rfloor}{NCP_{n+m-t-i}} \tag{19}$$

In the worst case, these sets of planes ( $p_1$, $p_2$, $p_3$ and $p_4$) are disjoint and one additional plane for a new connection $\langle x;Y;\omega \rangle$ is needed.

For case A, from (14), (16) and (17) we obtain:

$$p \geq \lfloor p_1 + p_2 + p_3 \rfloor + 1 \tag{20}$$

Next, for case B, from (14) and (16) we obtain:

$$p \geq \lfloor p_1 + p_2 \rfloor + 1 \tag{21}$$

For cases C and D, from (15), (16), (18) and (19) we obtain:

$$p \geq \lfloor p_1 + p_2 + p_3 + p_4 \rfloor + 1 \tag{22}$$

Equation (20), (21) and (22) must be maximized through all possible values of $\omega$ so we obtain following formula:

$$p \geq \max_{b \leq \omega \leq B} \{S(\omega)\} + 1 \tag{23}$$

where $S(\omega)$ is expressed by combination of $\lfloor p_1 + ... \rfloor$ related to (20-22) formulae.
After puting $\omega = B$ to formulae (20-22) we got (10-13) respectively.

## 4. Example

The example of the worst state of $\log_2(32;1;16)$ switching network, when $t = 2$, $\beta = 1$, $B = 0.6$, $b = 0.2$ (in case of $m = 1$, $t = \lfloor n/2 \rfloor$, $n$ is odd) is shown in Figure 1. In this example, each *BW* contains $2^2 = 4$ outputs, it follows that we have $2^5/2^2 = 8$ *BW*s. The considered new connection is $\langle 0;0;0.6 \rangle$. This connection of weight $0.6$ needs to be set up on input, output and all intermediate nodes. This connection can not be set up through a plane if other connections of total weight equal or greater than $1 - \omega + b = 0.6$ are already set up through any node on the path of this connection.

In stages from $n+m-t$ to $n+m-1$, we may set up connection from inputs 14-17 to outputs in the first *BW*. These connections we may set up through different planes and in the worst state they will block $p_1 = 2\ 1/2$ planes for $\langle 0;0;0.6 \rangle$. In stages from 1 to $t$ multicast connections may be set up. In the worst state these connections may be set up to each accessible blocking window (exception is the first blocking window where outputs are already occupied). There are connections of weight $b$ set up through stage 1. These connections are equivalent of two connections of weight $\alpha(\omega)$ (two connections

$\langle 0;\{4;8;12;18;20;24;28\};0.2\rangle$ and one connection $\langle 0;\{5;9;13;17;21;25;29\};0.2\rangle$ set up from input 0 and four connections $\langle 1;\{4;8;12;18;20;24;28\};0.2\rangle$ from input 1.

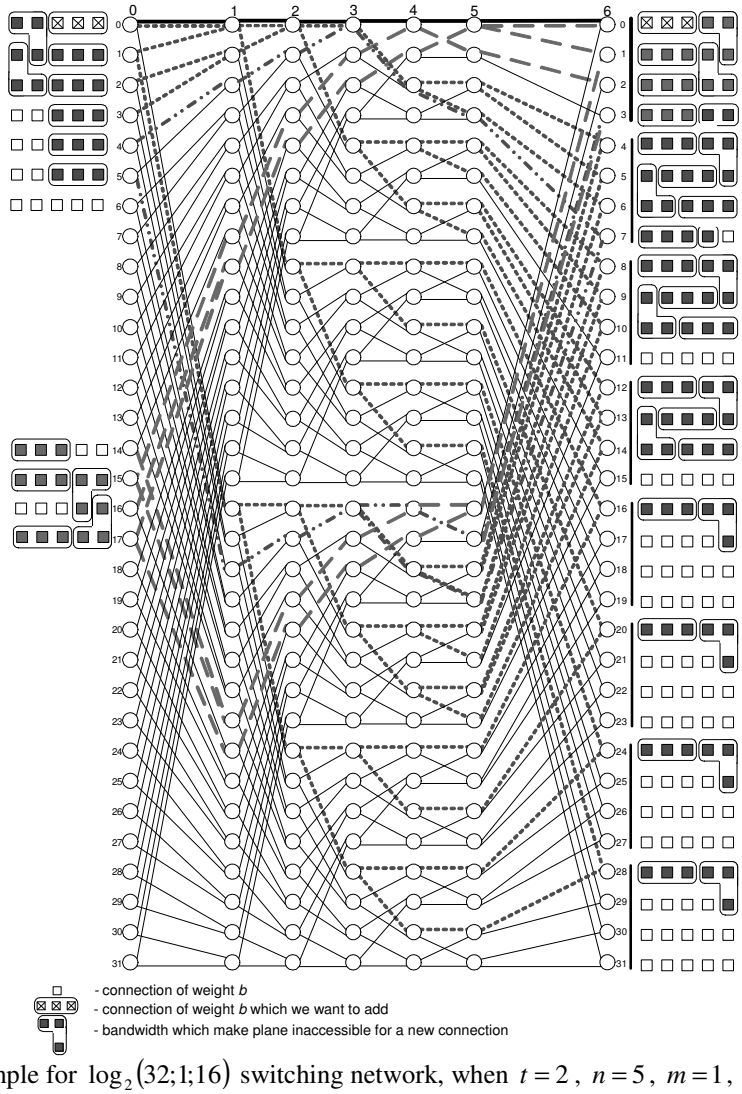



Fig.1. Example for $\log_2(32;1;16)$ switching network, when $t=2$, $n=5$, $m=1$, $\beta=1$, $\omega=0.6$, $b=0.2$.

Following connections are set up: one connection $\langle 1;\{5;9;13\};0.2\rangle$ and two connections $\langle 2;\{6;10;14\};0.2\rangle$, tree connections $\langle 2;\{5;9;13\};0.2\rangle$ and three connections $\langle 3;\{6;10;14\};0.2\rangle$ trough state 2. These connections together may block $p_2 = 11\ 1/2$ planes.

In stage $t+1$ connections may set up which may block paths in next planes. Number of these connections depends on a free bandwidth in accessible blocking windows from stage

$t+1$. In this example we have: three connections $\langle 4;\{7\};0.2\rangle$, two connections $\langle 5;\{3\};0.2\rangle$ and one connection $\langle 5;\{7\};0.2\rangle$. They may together block $p_3 = 1$ plane.

All of mentioned connections meet at one common node to $\langle 0;0;0.6\rangle$ at least. Therefore considered new connection has to be set up in $\lfloor p_1 + p_2 + p_3 \rfloor + 1 = 16$ plane.

## 5. Numerical results and analysis

In this section, we compare the result of different values of $\omega$ and $t$. In Figure 2 is shown dependence $\omega \in \{b; B\}$ on number planes. The first observation from this figure is that when $\omega$ increase then the number of planes increases as well. The explanation is that $S(\omega)$ is nondecreasing function and mostly is expressed in denominator. If $\omega$ increases, denominator decreases. The highest values of $S(\omega)$ is given for $\omega = B$.

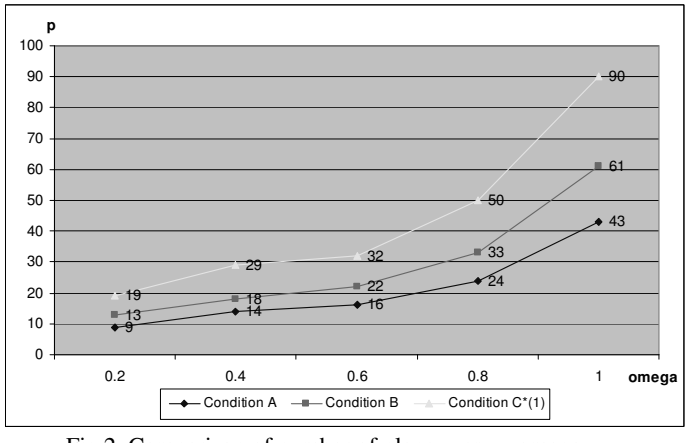



Fig.2. Comparison of number of planes versus omega.

We gave this results for condition A using following values: $n = 5$, $m = 1$, $t = 2$, $B = 1$, $\beta = 1$, $b = 0.2$ for condition B: $n = 6$, $m = 1$, $t = 3$, $B = 1$, $\beta = 1$, $b = 0.2$ for condition $C^{*(1)}$ $n = 6$, $m = 3$, $t = 2$, $B = 1$, $\beta = 1$, $b = 0.2$.

Analytical results of influence size of blocking window by changing $t$ value. On the figure 3 are shown that number of planes is inversely proportional of tendency to size of blocking window. Second observation from this picture is that for $t = 4$ and $t = 5$ we got the same number of planes. This connection of $t$, $n$ and $m$ planes which cause additional alternate paths make $p$ on the same level and $p = p_1$ for this case.

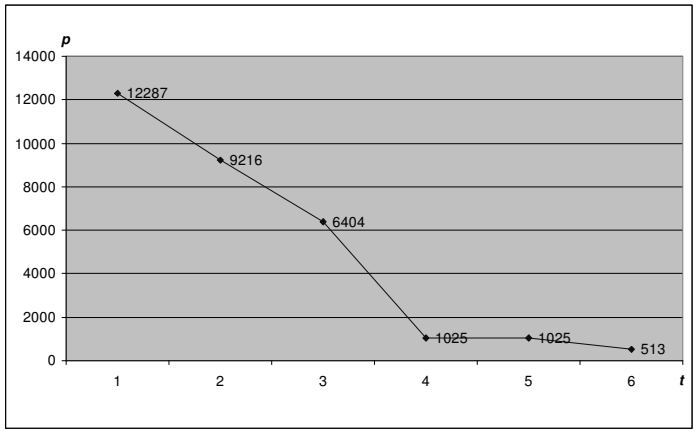



Fig.3. Comparison of number of planes versus size of blocking window.

We gave this results for condition A using following values: $n=12$, $m=4$, $B=1$, $\beta=1$, $b=0.125$.

## 6. Conclusions

In this paper we extend knows results for multi-log2($N$; $0$; $p$) to multi-rate multi-log2($N$; $m$; $p$) with multicast connections. We proved wide sense non-blocking conditions for discrete bandwidth case for $1 \le t \le \lfloor n/2 \rfloor$. These conditions are universal for both multirate and single-rate.

## 7. References

[1] G. Danilewicz G., Kabaciński W.: *Struktury s-sekcyjnych pól komutacyjnych zbudowanych z komutatorów λ*, KST 95 – conference materials, Bydgoszcz, September 1995, vol. E, pp. 157-166.

[2] Fujiwara M.: *A coherent photonic wavelength-division switching system for broadband networks*, J. Lightwave Technology, vol. 8, No. 3, 1990, pp. 416-422.

[3] Gerstel O.: *On the future of wavelength routing networks*, IEEE Networks, vol. 10, No. 6, 1996, pp.14-20.

[4] Hwang F.K., Jajszczyk A.: *Nonblocking multiconnection networks*, IEEE Transaction on Communications, Vol. COM-34, No. 10, 1986, pp. 1038-1041.

[5] Kaczmarek S.: *Własności równoległych połączeń pól komutacyjnych*, Przegląd Telekomunikacyjny, vol. LVI, No. 2, 1983, pp. 54-56.

[6] Lea C.-T.: *Multi-Log2N networks and their applications in high-speed electronic and photonic switching systems*, IEEE Transaction on Communication, vol. 38, No. 10 October 1990,pp. 1740-1749.

[7] Tscha Y., Lea K.H.: *Non-blocking conditions for multi-log2N multiconnection networks*, IEEE GLOBECOM 1992, pp.1600-1604

[8] Tscha Y., Lea K.H., *Yet another result on multi-log2N networks*, IEEE Transaction on Communication, vol. 47, No. 9, September 1999, pp.1600-1604.

[9] Danilewicz G., Kabaciński W.: *Wide-Sense Non-blocking Multi-Log2N Broadcast Switching Networks*, International Conference on Communications ICC 2000 New Orleans, LA USA, June 2000, pp. 1440-1444.

[10] Danilewicz G., Kabaciński W.: *Non-blocking multicast multi-log2N switching networks*, First Polish-German Teletraffic Symposium 2000, Drezno, September 2000, pp. 201-210.

[11] Danilewicz G., Kabaciński W.: *Wide-sense and strict-sense non-blocking operation of multicast multi-log2N switching networks*, IEEE Transactions on Communications, vol. 50, No. 6, June 2002, pp. 1025-1036.

[12] Danilewicz G., Kabaciński W.: *Comments "Wide-Sense Nonblocking Multicast Log2(N; m; p) Networks*, September 30, 2004.

[13] Frank K., Hwang and Bey-Chi Lin: *Wide-Sense Nonblocking Multicast Log2(N; m; p) Networks*, IEEE Transactions on Communications, VOL. 51, NO. 10, October 2003.

[14] Lea C.-T.: *Multirate Logd(N,e,p) Networks*, IEEE GLOBECOM 1994, pp.319-323.

[15] Kabaciński W., Żal M.: *Non-blocking operation of multi-Log2N switching networks*, System Science, vol. 25, No. 4, 1999, pp.83-97.

[16] Kabaciński W., Wichary T.: *Warunki nieblokowalności w polach typu multi-log$_2$N z połączeniami rozgłoszeniowymi typu multi-rate dla pasma dyskretnego*, PWT 2004, 9-10 XII

[17] Kabaciński W., Wichary T.: *Multi-log$_2$N Multirate Switching Networks with Multicast Connections,* 10$^{th}$ PSRT, Poland, 2003.

[18] Kabaciński W., Wichary T.: *Wide-Sense Non-blocking Multi-log$_2$N Multirate Switching Networks with Multicast Connections*, 12$^{th}$ PSRT, Poland, 2005.

[19] Suzu S.: *An experiment on high-speed optical time-division switching*, J. Lightwave Technology, vol. LT-4, No. 7, 1986, pp.894-899.

[20] Suzu S., Nagashima K.: *Optimal broadband communications network architecture utilizing wavelength-division switching technology*, Proc. First Topical Meeting on Photonic Switching, Incline Vilage, Nevada, 1987, pp. 134-137.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 65–76

# Point-to-Point Blocking Probability Calculation in Multi-service Switching Networks with BPP Traffic

MARIUSZ GŁĄBOWSKI [a]

[a]Chair of Communication and Computer Networks, Poznan University of Technology
mariusz.glabowski@et.put.poznan.pl

**Abstract:** This paper proposes the approximate calculation methods of the blocking probability in switching networks with point-to-point selection, which are offered multi-service traffic streams generated by Binomial (Engset)–Poisson (Erlang)-Pascal traffic sources. The proposed methods belong to the class of methods known as the *effective availability methods*. The basis of the proposed calculation algorithms is the occupancy distributions in interstage links as well as in the outgoing links (forming outgoing directions). These distributions are calculated with the help of the full-availability group model and the limited-availability group model. The results of analytical calculations of the blocking probabilities are compared with the simulation results of three-stage switching networks.

**Keywords:** BPP multi-rate traffic, switching networks, blocking probability

## 1. Introduction

Multi-service switching networks were the subject of many analyses [1–5]. The analytical methods of determination of traffic characteristics of such systems can be classified into two groups. In the first one time-effective algorithms of solving local balance equations in a multi-dimensional Markov process are searched for. However, in spite of its great accuracy, this method cannot be used for calculations of larger systems which have practical meaning. Methods of the other group consist in approximating a multi-dimensional service process by the appropriately constructed one-dimensional Markov chain, which is characterised by a product form solution [6–8]. Within the latter group, the most effective methods of switching networks calculations are the well-proven methods of the so-called *effective availability* [3, 9, 10]. The effective availability is defined as the availability in a multi-stage switching network in which the blocking probability is equal to the blocking probability of a single-stage network with the same capability of the outgoing group and at analogous parameters of the traffic stream offered. The modern methods of calculating the effective availability are based on works [9, 10] and [3], where all the components of this parameter have been defined. In [3], the practical and universal formulae for calculating the effective availability

have been derived for arbitrary multi-stage switching networks carrying a mixture of different multi-rate Poisson streams [3–5, 11].

Despite numerous studies in analytical modeling of switching networks with multi-rate traffic, in most of the published papers known to the author, only the switching networks with an infinite source population have been analysed. However, in modern networks, the ratio of source population and the capacity of a system is often limited and the value of traffic load offered by calls of particular classes is dependent on the number of occupied bandwidth units in the group, i.e. on the number of in-service traffic sources [12–15]. One of the most exemplifying up-to-date system of this kind is the UMTS system in which obtaining of predefined QoS parameters for particular services is accompanied with a necessity to limit the number of concurrent users serviced by a given NodeB. Simultanously, the switching techniques used in the UMTS system (and in ATM in particular) allow to determine equivalent bandwidths for particular classes of call streams and, in consequence, to apply multi-rate models. The first and only method of point-to-group blocking probability calculation in switching networks with a finite source population, limited to the Engset traffic case, was published in [16].

In this paper the new methods of blocking probability calculation in the switching networks with Binomial&Poisson&Pascal traffic streams has been proposed, i.e. Point-to-Point Blocking for Multi-rate traffic with Finite source population method (PPBMF) and Point-to-Point blocking for multi-rate traffic with Finite source population - Direct method (PPDF). The paper is organised as follows. Section 2 presents models of link group in switching networks servicing BPP multi-rate traffic streams. In Section 3, the PPBMF and PPDF methods of blocking probability calculation in multi-service switching network with BPP traffic are proposed. In Section 4, the calculation results are compared with the simulation results of 3-stage switching networks. Section 5 concludes the paper.

## 2. Links models in multiservice switching networks

### 2.1. Limited-availability group with an infinite population of traffic sources

Let us consider the limited-availability group (LAG) model, i.e. the system composed of $k$ separated transmission links. The system services call demands having an integer number of BBUs (Basic Bandwidth Units). Additionally, each of the links of the group has the capacity equal to $f$ BBUs. Thus, the total capacity of the system is equal to $V = kf$. The system services a call – only when this call can be entirely carried by the resources of an arbitrary single link. The group is offered $M$ independent classes of Poisson traffic streams having the intensities: $\lambda_1$,$\lambda_2$,...,$\lambda_M$. The holding time for calls of particular classes has an exponential distribution with the parameters: $\mu_1$,$\mu_2$,...,$\mu_M$. Thus, the mean traffic offered to the system by the class $i$ traffic stream is equal to $A_i = \lambda_i/\mu_i$. A class $i$ call requires $t_i$ BBUs to set up a connection. The occupancy distribution in the considered system can be determined on the basis of the generalized Kaufman-Roberts recursion (GKRR) [7, 8]:

$$n[P_n]_V = \sum_{i=1}^{M} A_i t_i \sigma_i \left(n - t_i\right) [P_{n-t_i}]_V, \tag{1}$$

where $[P_n]_V$ is the probability of an event in which there are $n$ busy BBUs in the system and $\sigma_i(n)$, the so-called *conditional propability of passing*, is the probability of admission of class $i$ call to the service when the system is found in the state $n$.

From Eq. (1) it results that the knowledge of the reverse transition rates $y_i(n)$ [17] of a class $i$ service stream outgoing from state $n$ is not required for the determination of the occupancy distribution in LAG with Poisson traffic streams. However, the value of this parameter is the basis of the method applied in this paper for the occupancy distribution calculation in the group with a finite population of traffic sources. The parameter $y_i(n)$ – which determines the average number of class $i$ calls serviced in the state $n+t_i$ – can be determined on the basis of the local balance equations in the considered group [17]:

$$y_i\left(n+t_i\right)=\begin{cases}A_i\sigma_i(n)[P_n]_V/[P_{n+t_i}]_V & \text{for } n+t_i\leq V,\\ 0 & \text{for } n+t_i>V.\end{cases} \tag{2}$$

### 2.2. Conditional Probability of Passing

The conditional probability of passing, that takes into account the dependence between call streams and the state of the system, determines part of the incoming call stream $\lambda_i$ to be transferred between the states $\{n\}$ and $\{n+t_i\}$ due to the specific structure of the limited-availability group. The parameter $\sigma_i(n)$ can be calculated as follows [3]: $\sigma_i(n)=1-(F(V-n,k,t_i-1,0)/F(V-n,k,f,0))$, where $F(x,k,f,t)$ is the number of arrangements of $x$ free BBUs in $k$ links, the capacity of each link is equal to $f$ BBUs and each link has at least $t$ free BBUs:

$$F(x,k,f,t)=\sum_{i=0}^{\left\lfloor\frac{x-kt}{f-t+1}\right\rfloor}(-1)^i\binom{k}{i}\binom{x-k(t-1)-1-i(f-t+1)}{k-1}. \tag{3}$$

Having the probabilities $\sigma_i(n)$, we can calculate the distribution $[P_n]_V$ and subsequently the blocking probability $e_i$ for class $i$ calls:

$$e_i=\sum\nolimits_{n=V-k(t_i-1)}^{V}[P_n]_V[1-\sigma_i(n)]. \tag{4}$$

### 2.3. Limited-availability group with finite and infinite population of traffic sources

This section proposes an approximate recursive algorithm that allows to determine the occupancy distribution in LAG with Binomial(Engset)&Poisson(Erlang)&Pascal traffic streams with different number of demanded BBUs. We start the present considerations with a short presentation of basic assumptions for the multi-service Engset and Pascal model.

#### 2.3.1. Assumptions of Engset Multi-rate Model

Let us consider now the system servicing multirate traffic generated by a finite population of sources. Let us denote as $N_j$ the number of sources of class $j$, the calls of which

require $t_j$ BBUs for service. The input traffic stream of class $j$ is built by the superposition of $N_j$ two-state traffic sources which can alternate between the active (busy) state ON (the source requires $t_j$ BBUs) and the inactive state OFF (the source is idle). When a source is busy, its call intensity is zero. Thus the arrival process is state-dependent. The class $j$ traffic offered by an idle source is equal to $\alpha_j = \Lambda_j/\mu_j$,where $\Lambda_j$ is the mean arrival rate generated by an idle source of class $j$ and $1/\mu_j$ is the mean holding (service) time of class $j$ calls. In the considered model, the holding time for calls of particular classes has an exponential distribution. Thus, the mean traffic offered to the system in the state of $n$ BBUs being busy by idle class $j$ traffic sources is equal to: $A_j(n) = (N_j - n_j(n))\alpha_j$, where $n_j(n)$ is the number of in-service sources of class $j$ in state $n$.

### 2.3.2. Assumptions of Pascal Multi-rate Model

Considering Pascal traffic streams we also assume a finite number of traffic sources. As in the Engset case, we assume that at the very beginning there are $S_k$ sources of class $k$ requiring $t_k$ BBUs. Each idle source generates calls with intensity $\gamma_k$. The holding time has an exponential distribution with the intensity $\mu_k$. Contrary to Engset Multi-rate Model, in the Pascal case, arrival intensity of particular traffic classes increases with the occupancy state of the system. This means that the arrival intensity of a class $k$ is equal $(S_k + n_k(n))\gamma_k$, where $n_k(n)$ is the number of in-service sources of class $k$ in state $n$. Thus, the mean traffic offered to the system in the state of $n$ BBUs being busy by class $k$ traffic sources is equal to $A_k(n) = (S_k + n_k(n))\beta_k$, where $\beta_k = \gamma_k/\mu_k$ is the mean traffic offered by an idle source of class $k$.

### 2.3.3. Concept of Determination of Multi-service Erlang-Engset-Pascal Distribution

Let us consider the limited-availability group with the capacity equal to $V$ BBUs which is offered three types of traffic streams: $M_1$ Erlang (Poisson) traffic streams, $M_2$ Engset (Binomial) traffic streams and $M_3$ Pascal traffic streams. As we can notice in Sections 2.3.1. and 2.3.2., the dependence of the value of the offered traffic on the number of active sources in Engset and Pascal streams makes it impossible to apply directly the GKRR. In this section we will discuss the idea of modeling multi-service switching networks by the application of the modified GKRR. The basis of this method is formed by a determination of an approximate method of determining the number of active traffic sources of a given class in Engset and Pascal streams.

Initially, we assume in the algorithm that the number of BBUs occupied in each of the states $n$ by respectively calls of class $j$ Engset stream and class $k$ Pascal stream, is the same as the number of BBUs occupied by the equivalent Erlang stream generating the offered traffic with the intensity:

$$A_j = N_j\alpha_j, \qquad A_k = S_k\beta_k, \tag{5}$$

which is equal in value to the traffic offered by all free sources of class $j$ Engset stream and class $k$ Pascal stream. The above adopted assumption also implies that the number of in-

service $n_j(n)$ and $n_k(n)$ sources of class $j$ and $k$, respectively, in the state of $n$ BBUs being busy, can be approximated by the reverse transition rates $y_j(n)$ and $y_k(n)$, determined on the basis of Equation (2) for the equivalent Erlang streams (Eq. (5)):

$$n_j(n) \sim y_j(n), \qquad n_k(n) \sim y_k(n). \tag{6}$$

The determined values of $y_j(n)$ and $y_k(n)$ enable us to make the mean value of offered traffic dependent on the occupancy state of the group in the following manner:

$$A_j(n) = [N_j - y_j(n)]\alpha_j, \qquad A_k(n) = [S_k + y_k(n)]\beta_k. \tag{7}$$

Having the traffic values $A_j(n)$ and $A_k(n)$, Equation (1) can be rewritten in the form that includes the traffic characteristics of Engset and Pascal traffic, namely:

$$n\,[P_n]_V = \sum_{i=1}^{M_1} A_i t_i \sigma_i(n-t_i)\,[P_{n-t_i}]_V + \sum_{j=1}^{M_2} A_j(n-t_j) t_j \sigma_j(n-t_j)\,\left[P_{n-t_j}\right]_V + \\ + \sum_{k=1}^{M_3} A_k(n-t_k) t_k \sigma_k(n-t_k)\,[P_{n-t_k}]_V. \tag{8}$$

### 2.4. Full-availability group

The full-availability group (FAG) is a discrete model of a single link that uses complete sharing policy [18]. This system is an example of a state-independent system in which the passage between two adjacent states of the process associated with a given class stream does not depend on the number of busy bandwidth units in the system. Therefore, the conditional state-passage probability $\sigma_i(n)$ in FAG is equal to 1 for all states and for each traffic class. Consequently, the occupancy distribution and blocking probabilities in the groups with infinite and finite source population can be calculated by the equations (8) and (4), taking into consideration the fact that: $\forall_i \forall_n \sigma_i(n) = 1$.

### 2.5. Distribution of available links

On the basis of the occupancy distribution in LAG (8), the so-called *distribution of available links* is determined [3]. This distribution determines the probability $P(i,s)$ of an event in which each of arbitrarily chosen $s$ links can carry the class $i$ call:

$$P(i,s) = \sum\nolimits_{n=0}^{V} [P_n]_V P(i,s|V-n), \tag{9}$$

where $[P_n]_V$ is the occupancy distribution in LAG with BPP traffic streams, and $P(i,s|x)$ the so-called *conditional distribution of available links* which determines the probability of an arrangement of $x$ ($x = V - n$) free BBUs, in which each of $s$ arbitrarily chosen links has

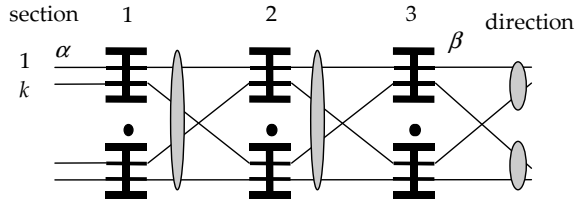



Fig. 1. A three stage switching network

at least $t_i$ free BBUs, while in each of the remaining $(k-s)$ links the number of free BBUs is lower than $t_i$ (Eq. (3)). Following the combinatorial consideration [3]:

$$P(i,s|x) = \binom{k}{s} \sum_{w=st_i}^{\Psi} F(w,s,f,t_i) F(x-w,k-s,t_i-1,0) \Big/ F(k,x,f,0), \quad (10)$$

where: $\Psi = sf$, if $x \geq sf$, $\quad \Psi = x$, if $x < sf$.

## 3. Switching network calculations

In this Section two approximate methods of point-to-point blocking probability calculation in multi-stage switching networks with multi-rate BPP traffic are presented, i.e. PPBMF (Point-to-Point Blocking for Multi-rate traffic with Finite source population) method and PPFD (Point-to-Point blocking for multi-rate traffic with Finite source population – Direct method) method. The presented considerations are based on PPBMT and PPD methods, worked out in [3] and [4] for switching networks with Poisson traffic streams.

The presented switching networks calculations are based on the reduction of calculations of internal blocking probability in a multi-stage switching network with BPP traffic to the calculation of the probability in an equivalent switching network model servicing single channel traffic. Such an approach allows us to analyse multi-stage switching networks with multi-rate traffic with the use of the effective availability method.

In the paper we consider the switching networks with the structure presented in Fig. 1. We assume that each of the inter-stage links has the capacity equal to $f$ BBUs and that outgoing transmission links create link groups called directions. Consequently, we assume that an interstage link can be modelled by the full-availability group and a direction can be modelled by the limited-availability group.

Furthermore, we consider switching networks with point-to-point selection [3]. The algorithm of setting up a connection in the switching network with point-to-point selection is aimed at finding a connection path between the outer-stage switches. If such a path does not exist, the connection is lost as the result of internal blocking. An external blocking event occurs when all links of a chosen direction are busy.

### 3.1. PPBMF Method

Let us consider now the PPBMF method for blocking probability calculation in switching networks with point-to-point selection, servicing multi-rate BPP traffic streams. The basis for the proposed method is the PPBMT method (Point-to-Point Blocking for Multi-channel Traffic) worked out in [3] for switching networks with infinite source populations. Modifications to the PPBMF method consists in the introduction of the appropriate group models with traffic generated by the finite source population, determined in Sect. 2., to calculations. In the method, blocking probability calculations for the switching networks are made in accordance with Lotze's remark [9] that point-to-point blocking in $z$-stage switching network is equal to point-to-group blocking in a $(z-1)$-stage switching network.

Let us assume that a certain switch $\beta$ belonging to the last stage of the switching network, chosen by the control system, has $t_i$ free BBUs necessary to set up the class $i$ connection. We can also assume that for the switch $\alpha$ (on the incoming links of which there appears the class $i$ call) there are $d_e(i)$ available interstage links coming to the destination switch $\beta$ from the last but one stage. The internal point-to-point blocking phenomenon appears when none of the $d_e(i)$ available links have a sufficient number of BBUs for servicing the class $i$ call:

$$E_i^{in} = \sum_{s=0}^{k-d_e(i)} P(i, s \wedge 1) \left[ \binom{k-s}{d_e(i)} / \binom{k}{d_e(i)} \right], \tag{11}$$

where $P(i, s \wedge 1)$ is the so-called *combinatorial distribution of available links in a switch*.

The probability of the internal point-to-point blocking for the class $i$ call stream is calculated with the assumption that at least one incoming link and one outgoing link of the system have at least $t_i$ free BBUs. The fact that one of the incoming links of the switch is available for the class $i$ call does not mean simultaneously that one of its outgoing links is also available [3]. The probability of available links in a switch $P(i, s \wedge 1)$ was determined on the basis of conditional distribution (10) of available links in the limited-availability group. This probability determines an event in which $s$ incoming links and, at the same time, at least one of the outgoing links of a given switch (e.g. the switch $\beta$) are available for the class $i$ call. According to the consideration worked out in [3], this distribution can be written as follows:

$$P(i, s \wedge 1) = \frac{\sum_{x=0}^{V} P(i, s|x)[1 - P(i, 0|x)][P_{V-x}]_V}{1 - \sum_{n=0}^{k} \left[ \sum_{x=0}^{V} P(i, n|x) P(i, 0|x) [P_{V-x}]_V \right]}, \tag{12}$$

where $P(i, s|x)$ – conditional distribution of available links in LAG with BPP traffic, $[P_n]_V$ – occupancy distribution in LAG with BPP traffic.

The phenomenon of the external blocking occurs when none of outgoing links of the demanded direction of the switching network can service the class $i$ call (i.e. does not have $t_i$ free BBUs). The occupancy distribution of the outgoing direction can be approximated by the distribution of available links in LAG with BPP traffic. Thus, the external blocking probability can be calculated by the formula:

$$E_i^{ex} = P(i, 0), \tag{13}$$

where $P(i,s)$ is the distribution of available links for class $i$ calls in LAG with BPP traffic.

The total blocking probability $E_i$ for the class $i$ call is a sum of external and internal blocking probabilities. Assuming the independence of internal and external blocking events, we obtain:

$$E_i = E_i^{ex} + E_i^{in}[1 - E_i^{ex}]. \tag{14}$$

For blocking probability calculation $E_i$, it is necessary to determine the value of $d(i)$. The parameter $d(i)$ is known as the effective availability of the switching network for the class $i$ call stream and will be described in Sect. 3.3..

### 3.2. PPFD method

In the other of the proposed method of blocking probability calculation in switching networks with point-to-point selection and finite population of traffic sources – the PPDF method – the evaluation of the internal point-to-point blocking probability is made on the basis of the effective availability quotient and the capacity of an outgoing group. The proposed method is based on the the PPD method, elaborated in [4] for switching networks with infinite source population.

In order to explain the basic assumptions of the proposed method, let us consider a switching network with point-to-point selection. An outgoing link belonging to a given last-stage switch is considered to be available for the first-stage switch if it is possible to set up a class *i* connection between these switches. Let us assume that the $z$-stage switching network is in a state $X$. The control system determines the first-stage switch, on the incoming link of which there appears a class $i$ call (switch $\alpha$). First, the control system finds the last-stage switch (switch $\beta$) having a free outgoing link in the demanded direction. Then, the control system tries to find a connection path for class $i$ call between the switches $\alpha$ and $\beta$ . Let us assume that in the state $X$ there are $d(i,X)$ available last-stage switches for the switch $\alpha$. If the chosen switch $\beta$ belongs to the group of $d(i,X)$ available switches, then class $i$ connection is set up, otherwise connection is lost because of the internal blocking event. Thus, the probability of the internal blocking can be determined as a ratio of free links belonging to an unavailable group of $V - d(i,X)$ switches to all free links in a given direction. If we assume that the probability of links occupancy is the same for all links of the direction, the average value of internal blocking is equal to:

$$E_i^{in} = \sum\nolimits_{\Omega}[(V - d(i,X))/V]P(X), \tag{15}$$

where: $\Omega$ - is the set of all possible states $X$ of a switching network, and $P(X)$ - is the state probability of switching network. If we designate the average value of available last-stage switches by $d_e(i)$, Equation (15) can be finally rewritten as follows:

$$E_i^{in} = [V - d_e(i)]/V. \tag{16}$$

The phenomenon of external blocking occurs when none of outgoing links of the demanded direction in a switching network can service a class $i$ call. The occupancy distri-

bution of the outgoing direction can be approximated by the occupancy distribution in the limited availability group. Consequently, the external blocking probability $E_i^{ex}$ and the total blocking probability $E_i$ for class $i$ calls, can be calculated by (13) and (14), respectively

### 3.3. Effective availability

The concept of the so-called equivalent switching network [3] is the base for effective availability calculation for class $i$ traffic stream. Following this concept, the network with multi-rate traffic is reduced to an equivalent network carrying a single-rate traffic. Each link of the equivalent network is treated as a single-channel link with a fictitious load $e_l(i)$ equal to blocking probability for a class $i$ stream in a link of a real switching network between section $l$ and $l+1$. This probability can be calculated on the basis of the occupancy distribution in the full-availability group with BPP traffic streams (Sect. 2.4.). The effective availability in a real $z$-stage switching network is equal to the effective availability in an equivalent switching network and can be determined by the formula derived in [3]:

$$d(i) = [1 - \pi_z(i)]k + \pi_z(i)\eta Y_1(i) + \pi_z(i)[k - \eta Y_1(i)]e_z(i)\sigma_z(i), \tag{17}$$

where:

- $d(i)$ – the effective availability for the class $i$ traffic stream in an equivalent network,

  $\pi_z(i)$ – the probability of an event where the class $i$ connection path cannot be set up between a given first-stage switch and a given last-stage switch.
- $k$ – the number of outgoing links from the first stage switch,
- $Y_1(i) = ke_1(i)$ – the fictitious traffic served by the switch of the first stage:
- $e_c(i)$ – the blocking probability for the class $i$ stream in an interstage link (between stages $c$ and $c+1$) of a real network. The $e_c(i)$ parameter can be calculated on the basis of the full-availability group model with multi-rate traffic (Sect. 2.4.).
- $\eta$ – a portion of the average fictitious traffic from the switch of the first stage which is carried by the direction in question. If the traffic is uniformly distributed between all $h$ directions, we obtain $\eta = 1/h$,
- $\sigma_z(i)$ – the so-called *secondary availability coefficient* [3]:

$$\sigma_z(i) = 1 - \prod_{j=2}^{z-1} \pi_j(i) \tag{18}$$

## 4. Calculation and simulation results

In order to confirm the adopted assumptions in the PPBMF and PPFD method, the results of the analytical calculations were compared with the simulation results of a 3-stage switching network. The structure of the switching network consisting of the switches of

$k \times k$ links is shown in Fig. 1. The results presented in the paper (Figs. 2(a)–2(f)) were obtained for the switching network with the parameters: $k = 4$, $f = 30$, $t_1 = 1$, $t_2 = 2$, $t_3 = 6$. The research was carried out for different distinct values of the ratio of the number of traffic sources (Pascal and Engset traffic streams) of all classes and the switching network capacity. The results of the simulation are shown in the charts in the form of marks with 95% confidence intervals that have been calculated after the $t$–Student distribution for the five series with 1,000,000 calls of this traffic class that generates the lowest number of calls. All the results are expressed in relation to the value of total traffic $a$ offered to a single BBU at the entry to the network. Figures 2(a) and 2(b) show the results of point-to-point blocking probability in the switching network with an infinite source population. The results obtained allow us to compare the accuracy of the model of the switching network with an infinite population of traffic sources with the accuracy of the proposed calculation method in the case of the switching network with a finite source population, both for Engset (Figs. 2(c), 2(d)) and Pascal traffic streams (Figs. 2(e), 2(f)).

## 5. Conclusions

The paper proposes the approximate methods of point-to-point blocking probability calculation in switching networks with multi-rate traffic generated by a finite source population (BPP traffic). The method is based on the concept of effective availability. The analytical results of blocking probability, obtained on the basis of the proposed method, are compared with the simulation results. The simulation results confirm high accuracy of the proposed analytical model. Due to the limited space available in the paper, we have restricted ourselves to present only the selected results. However, numerous simulation experiments indicate that similar accuracy of the proposed analytical model can be obtained for various structures of switching networks and for various number of traffic classes.

## References

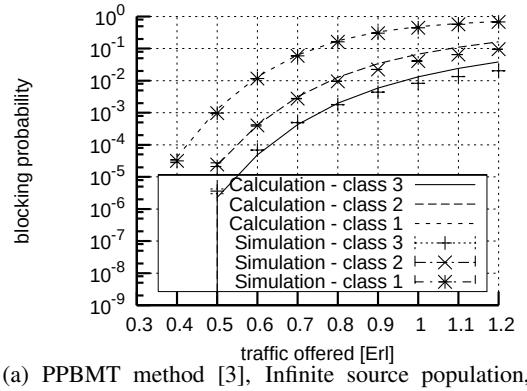
[1] J. Conradt and A. Buchheister, "Considerations on loss probability of multi-slot connections," in *Proc. 11th ITC*, Kyoto, 1985, pp. 4.4B–2.1.

[2] M. Beshai and D. Manfield, "Multichannel services performance of switching networks," in *Proc. 12th ITC*. Torino:, 1988, pp. 857–864.

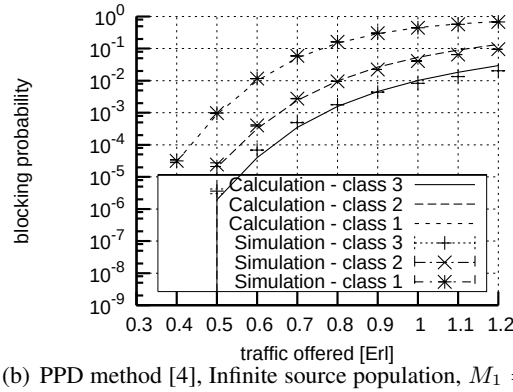
[3] M. Stasiak, "Combinatorial considerations for switching systems carrying multichannel traffic streams," *Ann. des Télécomm.*, vol. 51, no. 11–12, pp. 611–625, 1996.

[4] M. Stasiak and M. Głąbowski, "Point-to-point blocking probability in switching networks with reservation," in *Proc. 16th ITC*, vol. 3A. Edinburgh, 1999, pp. 519–528.

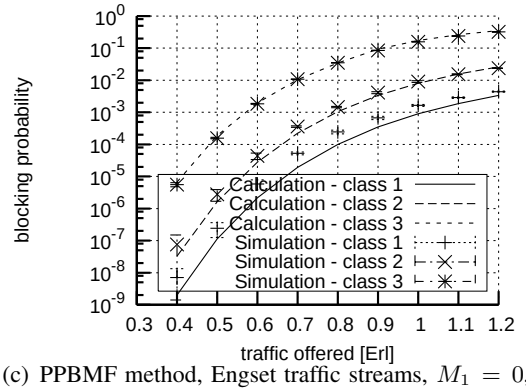
[5] M. Głąbowski and M. Stasiak, "Point-to-point blocking probability in switching networks with reservation," *Ann. des Télécomm.*, vol. 57, no. 7–8, pp. 798–831, 2002.



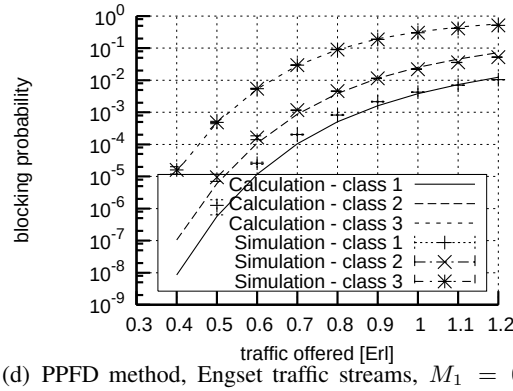

(a) PPBMT method [3], Infinite source population, $M_1 = 3$, $M_2 = 0$, $M_3 = 0$



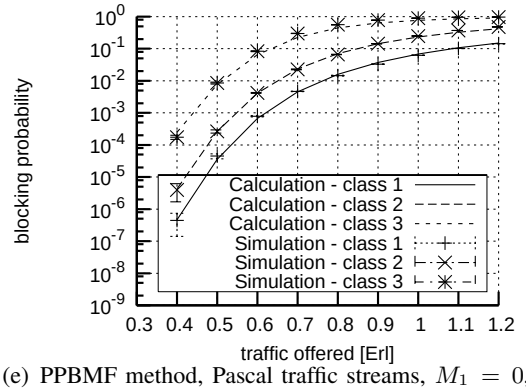

(b) PPD method [4], Infinite source population, $M_1 = 3$, $M_2 = 0$, $M_3 = 0$



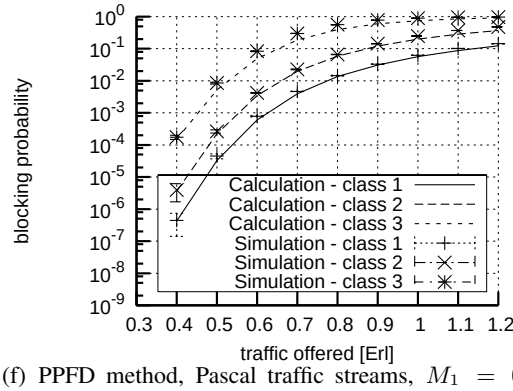

(c) PPBMF method, Engset traffic streams, $M_1 = 0$, $M_2 = 3$, $M_3 = 0$, $N_1 = N_2 = N_3 = 320$




(d) PPFD method, Engset traffic streams, $M_1 = 0$, $M_2 = 3$, $M_3 = 0$, $N_1 = N_2 = N_3 = 800$




(e) PPBMF method, Pascal traffic streams, $M_1 = 0$, $M_2 = 0$, $M_3 = 3$, $S_1 = S_2 = S_3 = 320$




(f) PPFD method, Pascal traffic streams, $M_1 = 0$, $M_2 = 0$, $M_3 = 3$, $S_1 = S_2 = S_3 = 320$

Fig. 2. Point-to-point blocking probability in the switching network, three traffic classes, $A_1t_1 : A_2t_2 : A_3t_3 = 1 : 1 : 1$

[6] V. Iversen, "The exact evaluation of multi-service loss systems with access control," in *Seventh Nordic Teletraffic Seminar (NTS-7)*, Lund,Sweden, Aug. 1987, pp. 56–61.

[7] J. Kaufman, "Blocking in a shared resource environment," *IEEE Transactions on Communications*, vol. 29, no. 10, pp. 1474–1481, 1981.

[8] J. Roberts, "A service system with heterogeneous user requirements — application to multi-service telecommunications systems," in *Proc. of Performance of Data Communications Systems and their Applications*, Amsterdam, 1981, pp. 423–431.

[9] A. Lotze, A. Roder, and G. Thierer, "PPL — a reliable method for the calculation of point-to-point loss in link systems," in *Proc. 8th ITC*, Melbourne, 1976, pp. 547/1–44.

[10] E. Ershova and V. Ershov, *Digital Systems for Information Distribution*. Moscow: Radio and Communications, 1983, in Russian.

[11] M. Stasiak and M. Głąbowski, "PPBMR method of blocking probability calculation in switching networks with reservation," in *Proc. GLOBECOM 1999*, vol. 1a, Rio de Janeiro, 1999, pp. 32–36.

[12] Y. Kogan and M. Shenfild, "Asymptotic solution of generalized multiclass Engset model," in *Proc. 14th ITC*, Antibes Juan-les-Pins, France: Elsevier, 1994, pp. 1239–1249.

[13] G. Choudhury, K. Leung, and W. Whitt, "An inversion algorithm to compute blocking probabilities in loss networks with state-dependent rates," *IEEE/ACM Trans. Networking*, vol. 3, pp. 585–601, 1995.

[14] S. Berezner and A. Krzesinski, "An efficient stable recursion to compute multiservice blocking probabilities," *Journal of Performance Evaluation*, vol. 43, no. 2–3, pp. 151–164, 2001.

[15] M. Ermel and et al., "Performance of GSM networks with general packet radio services," *Journal of Performance Evaluation*, vol. 48, no. 1–4, pp. 285–310, 2002.

[16] M. Głąbowski, "Blocking probability in multi-service switching networks with finite source population," in *Proc. 14th IEEE International Conference On Telecommunications*, Penang, Malaysia, may 2007.

[17] M. Stasiak and M. Głąbowski, "A simple approximation of the link model with reservation by a one-dimensional Markov chain," *Journal of Performance Evaluation*, vol. 41, no. 2–3, pp. 195–208, Jul. 2000.

[18] J. Roberts, V. Mocci, and I. Virtamo, Eds., *Broadband Network Teletraffic, Final Report of Action COST 242*. Berlin: Commission of the European Communities, Springer, 1996.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 77–88

# Iterative Algorithm for Blocking Probability Calculation in Erlang-Engset-Pascal Multi-rate Systems

MARIUSZ GŁĄBOWSKI(1) ADAM KALISZAN(2)
MACIEJ STASIAK(3)

Chair of Communication and Computer Networks,
Poznan University of Technology
(1) *mariusz.glabowski@et.put.poznan.pl*
(2) *adam.kaliszan@gmail.com*
(3) *stasiak@et.put.poznan.pl*

**Abstract:** This paper proposes an approximate calculation method of occupancy distribution and blocking probability in systems which are offered multi-service traffic streams generated by Binomial–Poisson–Pascal (BPP) traffic sources. The method is based on transforming a multi-dimensional service process in the system into a one-dimensional Markov chain and on appropriate modification of the generalized Kaufman-Roberts recursion. The proposed algorithm determines the number of sources of particular classes, being serviced in a given state of the system, and subsequently the traffic offered in particular states of the system occupancy. The results of analytical calculations of the blocking probabilities in exemplary systems with BPP traffic streams, obtained on the basis of the proposed method, are compared with the results of the exact Iversen convolution algorithm.

**Keywords:** BPP multi-rate traffic, blocking probability

## 1. Introduction

The analytical methods for determination of traffic characteristics of systems with multi-service traffic can be classified into three groups. In the first one, time-effective algorithms of solving local balance equations in a multidimensional Markov service process, occurring in the switching systems, are searched for [1–7]. These methods can be theoretically used to model any communication system with arbitrary traffic streams. However, in practice, these methods cannot be used for calculations of large communication systems because of an excessive number of states in which a multi-dimensional Markov process occurring within the system can take place [8,9]. In the second group there are considered the methods based on the Iversen convolution algorithm [1, 5, 10]. This algorithm allows us to calculate the systems with state-independent service process [1] and with Binomial (Engset)–Poisson

[1] In the teletraffic theory, the term "state-dependent system" is usually used. However, it should be noticed that the "state-dependence" can result from the specific feature of the servicing system (e.g. call admission policy) or

(Erlang)–Pascal traffic streams [10]. Unfortunately, only the so-called full-availability group (FAG) can be treated as the system with state-independent service process. Introduction of any limitation in accepting new calls causes a communication systems to become the system dependent on a state, i.e. the system in which the admission of a new call is conditioned not only by the sufficient number of free BBUs (Basic Bandwidth Units) to service a given class call, but also by the current state of the occupancy distribution of the system.

The methods of the third group is based on the approximation of a multi-dimensional service process, occurring in the communication systems, by the one-dimensional Markov chain [11–14]. In [15, 16] it was proved that a multi-dimensional service process can be reduced accurately to the one-dimensional Markov chain in the case of the FAG with multi-rate Poisson traffic. Such a reduction forms the basis for the determination of the occupancy distribution in the system by means of the recurrent Fortet-Grandjean formula [17] which is generally known as the Kaufman-Roberts recursion [15, 16]. Further generalization of the Kaufman-Roberts recursion for the systems with state-dependent service process is based on the introduction of relevant conditional state-passage coefficients between the adjacent states of the system [11, 13, 18]. This approach allows us to model these systems by means of a simple recurrent formula. However, due to the limitations the Kaufman-Roberts recursion imposes, as far as the type of offered traffic is concerned (only Poisson traffic), it is not possible to apply this recursion directly to the state-dependent system with Binomial and Pascal traffic streams. It should be stressed that for the FAG with BPP multi-rate traffic there is a precise solution proposed in [14]. This solution, however, cannot be applied to the generalized case, i.e. for systems with state-dependent service process, in which call streams carried between neighboring states associated with a given class stream depend on the current state of the process. Consequently, in [19, 20] a simple calculational occupancy distribution and blocking probability algorithm, based on the Kaufman-Roberts recursion, is proposed in the so-called Engset Multi-Rate Loss Model (EnMLM), i.e. in the model with truncated Binomial distribution, where capacity of a system is lower than the number of sources. The basis of the proposed method [19] is formed by an approximate algorithm of determining the number of active traffic sources of a given class in Engset streams. However, the method is based on a single iteration. It may cause some inaccuracies for small number of traffic sources and lead to impossibility of determining the accuracy of the algorithm.

Therefore, in the present paper we propose the new method of the occupancy distribution calculation in multi-service systems with finite source population. This method is based on multiple iterations and allows us to set the accuracy of the iterative process. Additionally, a coherent methodology for determining traffic characteristics of multi-service systems that simultaneously service Binomial(Engset)-Poisson(Erlang)-Pascal call streams is proposed in the article. In the works published so far, only those systems have been considered that are offered simultaneously traffic of one type, i.e. either Erlang multi-rate traffic (with Poisson

from the specific feature of traffic sources (e.g. in the case of limited number of traffic sources). Therefore, in the present paper we used the terms "system with state-dependent service process" and "system with state-dependent arrival process" in order to distinguish both the indicated phenomena.

distribution), or Engset multi-rate traffic (with Binomial distribution).

The remaining part of the paper is organized as follows. Section 2 describes the generalized Kaufman-Roberts recursion for the Erlang multi-rate traffic streams. In Section 3, our methodology of modeling the systems with multi-rate BPP traffic is proposed. In Section 4, accuracy of the proposed method is evaluated. Section 5 concludes the paper.

## 2. Full-Availability Group with Infinite Population of Traffic Sources

Let us consider the full-availability group (FAG) with different multi-rate traffic streams. The FAG is a discrete link model that uses complete sharing policy [21]. Let us assume that the system services call demands having an integer number of the so-called BBUs (Basic Bandwidth Units)[2]. The total capacity of the group is equal to $V$ BBUs. The group is offered $M_1$ independent classes of Poisson traffic streams having the intensities: $\lambda_1$, $\lambda_2$,...,$\lambda_{M_1}$. A class $i$ call requires $t_i$ basic bandwidth units. The holding time for calls of particular classes has an exponential distribution with the parameters: $\mu_1$, $\mu_2$,...,$\mu_{M_1}$. Thus the mean traffic offered to the system by the class $i$ traffic stream is equal to:

$$A_i = \lambda_i/\mu_i. \tag{1}$$

The occupancy distribution in the FAG can be determined on the basis of the generalized Kaufman-Roberts recursion (GKRR) [15, 16]:

$$n\,[P_n]_V = \sum\nolimits_{i=1}^{M_1} A_i t_i \sigma_{i,S}(n-t_i)\,[P_{n-t_i}]_V, \tag{2}$$

where $[P_{n-t_i}]_V$ is the occupancy distribution, i.e. the probability of an event if there are $n$ busy BBUs in the system, and $\sigma_{i,S}(n)$ is the state-passage coefficient in the system with state-dependent service process.

The FAG is an example of a system with state-independent service process, in which the passage between two adjacent states of the process associated with a given class stream does not depend on the number of busy bandwidth units in the system. Therefore, $\sigma_{i,S}(n)$ in the FAG is equal to 1 for all states and for each traffic class: $\forall_{1\le i\le M_1}\forall_{0\le n\le V}\,\sigma_{i,S}(n)=1$.

The blocking state for class $i$ calls occurs in the FAG when the group does not have $t_i$ free BBUs which are required for servicing calls of this traffic class. Thus, the blocking probability for class $i$ calls can be calculated by the formula:

$$E_i = \sum\nolimits_{n=V-t_i+1}^{V} [P_n]_V. \tag{3}$$

Figure 1 shows a graphical representation of the distribution (2) for a system with two call streams. The call of the first class demands $t_1 = 1$ BBU to set up a connection, while the call of the second class $t_2 = 2$ BBUs. The $y_i(n)$ symbol denotes reverse transition rates of

[2]While constructing multi-rate models for broadband systems, it is assumed that BBU is the greatest common divisor of equivalent bandwidths of all call streams offered to a system [21, 22].

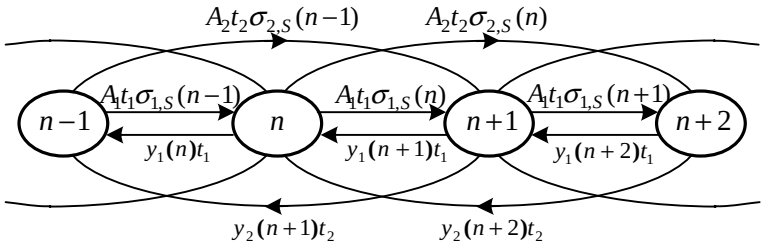



Fig. 1. A fragment of one-dimensional Markov chain in a state-dependent system with two call streams ($t_1 = 1, t_2 = 2$)

a class $i$ service stream outgoing from state $n$. These transition rates for a class $i$ stream are equal to the average number of class $i$ calls serviced in state $n$. From Eq. (2) it results that the knowledge of the parameter $y_i(n)$ is not required for determining the occupancy distribution in systems with traffic generated by an infinite population of traffic sources (Poisson traffic). However, the value of parameter $y_i(n)$, in a given state of the group is the basis of the method (proposed in the further part of the paper) of occupancy distribution calculation in the group with a finite population of traffic sources (Binomial and Pascal traffic). The parameter $y_i(n)$ can be calculated on the basis of the local balance equations [23]:

$$y_i(n) = \begin{cases} A_i \sigma_{i,S}(n - t_i) \left[P_{n-t_i}\right]_V / \left[P_n\right]_V & \text{for } n \leq V, \\ 0 & \text{for } n > V. \end{cases} \tag{4}$$

## 3. Full-Availability Group with BPP Traffic

This section proposes an algorithm with multiple iterations, that allows to determine the occupancy distribution in the FAG with BPP traffic streams with different number of demanded BBUs. We start the considerations with a short presentation of basic assumptions for the multi-service Engset and Pascal models.

### 3.1. Assumptions of Engset Multi-rate Model

Let us consider now the system servicing multirate traffic generated by a finite population of sources. Let us denote as $N_j$ the number of sources of class $j$[3], the calls of which require $t_j$ BBUs for service. The input traffic stream of class $j$ is built by the superposition of $N_j$ two-state traffic sources which can alternate between the active (busy) state ON (the source requires $t_j$ BBUs) and the inactive state OFF (the source is idle). When a source is busy, its call intensity is zero. Thus the arrival process is state-dependent. The class $j$ traffic offered by an idle source is equal to $\alpha_j = \Lambda_j/\mu_j$,where $\Lambda_j$ is the mean arrival rate generated by an idle source of class $j$ and $1/\mu_j$ is the mean holding (service) time of class $j$ calls. In

[3]In the present paper, the letter "$i$" denotes Erlang traffic class, the letter "$j$" — Engset traffic class, and the letter "$k$" — Pascal traffic class.

the considered model, the holding time for calls of particular classes has an exponential distribution. Thus, the mean traffic offered to the system in the state of $n$ BBUs being busy by idle class $j$ traffic sources is equal to: $A_j(n) = (N_j - n_j(n))\alpha_j$, where $n_j(n)$ is the number of in-service sources of class $j$ in state $n$.

### 3.2. Assumptions of Pascal Multi-rate Model

Considering Pascal traffic streams we also assume a finite number of traffic sources. As in the Engset case, we assume that at the very beginning there are $S_k$ sources of class $k$ requiring $t_k$ BBUs. Each idle source generates calls with intensity $\gamma_k$. The holding time has an exponential distribution with the intensity $\mu_k$. Contrary to Engset Multi-rate Model, in the Pascal case, arrival intensity of particular traffic classes increases with the occupancy state of the system. This means that the arrival intensity of a class $k$ is equal $(S_k + n_k(n))\gamma_k$, where $n_k(n)$ is the number of in-service sources of class $k$ in state $n$. Thus, the mean traffic offered to the system in the state of $n$ BBUs being busy by class $k$ traffic sources is equal to $A_k(n) = (S_k + n_k(n))\beta_k$, where $\beta_k = \gamma_k/\mu_k$ is the mean traffic offered by an idle source of class $k$.

### 3.3. Single Iteration Method – SIM

Interrelation between the offered traffic and the number of in-service sources in Engset Multi-rate Model makes the direct application of the GKRR (2) for determining the occupancy distribution in the considered system impossible. In [19] an approximate method has been proposed, which enables us to make the mean value of traffic offered by class $j$ dependent on the occupancy state (the number of occupied BBUs) of the group, and thereby determination of the system with a finite population of sources by the GKRR. In the considered method it is assumed that the number of in-service $n_j$ sources of class $j$ in the state of $n$ BBUs being busy is approximated by the parameter $y_j(n)$ which determines the average number of class $j$ calls serviced in the state of $n$ busy BBUs:

$$n_j(n) \sim y_j(n). \tag{5}$$

With such an approach it is assumed that the average number of given class calls, being serviced in the given state of the group with an infinite population of traffic sources, is approximate to the average number of calls being serviced in the same state in the case of a finite source population. Thereby, parameter $y_j(n)$ can be determined by Eq. (4), where probabilities $[P_n]_V$ are calculated by the GKRR, under the initial assumption that the offered traffic is not dependent on the number of in-service sources and equal to:

$$A_j = N_j\alpha_j. \tag{6}$$

The determined values of $y_j(n)$ enables us to make the mean value of offered traffic dependent on the occupancy state of the group in the following manner:

$$A_j(n) = (N_j - y_j(n))\alpha_j. \tag{7}$$

Eventually, the occupancy distribution in the FAG with multi-rate Engset traffic streams can be approximated by the formula:

$$n\,[P_n]_V = \sum_{j=1}^{M_2} A_j(n-t_j)t_j\left[P_{n-t_j}\right]_V, \tag{8}$$

where $M_2$ is the number of Engset traffic classes.

On the basis of the above equations, the algorithm of blocking probability calculations in the multi-rate group with a finite population of traffic sources may be written as follows:

1. Calculation of offered traffic $A_j$ of class $j$ on the basis of Eq. (6).
2. Determination of state probabilities $[P_n]_V$ – for the known value of offered traffic $A_j$ – in the FAG with an infinite population of traffic sources (Eq. (2)).
3. Calculation of reverse transition rates $y_j(n)$ on the basis of Eq. (4). Parameter $y_j(n)$ approximates the average value of in-service sources of class $j$ in the state of $n$ busy BBUs in the group with a finite source population.
4. On the basis of Eq. (7) the value of offered traffic is made dependent on the occupancy state of the group and subsequently the occupancy distribution in the group with a finite source population is determined according to Eq. (8).
5. Determination of blocking probabilities for class $j$ calls (Eq. (3)).

The main disadvantage of the presented algorithm is the determination of the number of traffic sources, being serviced in a given state of the system, on the basis of the occupancy distribution obtained for the equivalent Erlang streams (Step 2). Such an approach can lead to overestimate the real number of Engset traffic sources. In order to increase the accuracy of the presented Single Iteration Method, in the further part of the paper we propose a new Multiple Iteration Method (MIM). In the MIM, the number of active sources in a given state of the system is determined in a several iterations, with the set value of the relative error of the iterative process.

### 3.4. Multiple Iteration Method – MIM

Let us consider the FAG with the capacity equal to $V$ BBUs which is offered three types of traffic streams: $M_1$ Erlang (Poisson) traffic streams, $M_2$ Engset (Binomial) traffic streams and $M_3$ Pascal traffic streams. As we can notice in Sections 3.1. and 3.2., the dependence of the value of the offered traffic on the number of active sources in Engset and Pascal streams makes it impossible to apply directly the GKRR. In this section we will discuss the idea of modeling multi-service systems by the application of the modified GKRR applying multiple iterations for determining the number of active sources, i.e. the so-called Multiple Iterations Method. The basis of MIM, analogically as in the case of the method with single iteration, is formed by an approximate method of determining the number of active traffic sources of a given class in Engset and Pascal streams.

Initially, as in the case of SIM, we assume in the algorithm that the number of BBUs occupied in each of the states $n$ by calls of class $j$ Engset stream and class $k$ Pascal stream respectively, is the same as the number of BBUs occupied by the equivalent Erlang stream generating the offered traffic with the intensity:

$$A_j = N_j \alpha_j, \qquad A_k = S_k \beta_k, \tag{9}$$

which is equal in value to the traffic offered by all free sources of class $j$ Engset stream and class $k$ Pascal stream. The above adopted assumption also implies that the number of in-service $n_j(n)$ and $n_k(n)$ sources of class $j$ and $k$, respectively, in the state of $n$ BBUs being busy, can be approximated by the reverse transition rates $y_j(n)$ and $y_k(n)$, determined on the basis of Eq. (2) for the equivalent Erlang streams (Eq. (9)):

$$n_j(n) \sim y_j(n), \qquad n_k(n) \sim y_k(n). \tag{10}$$

In order to emphasize the dependence of Engset and Pascal traffic streams on the occupancy state of the system, let us express the value of offered traffic in particular states of the system by the state-passage coefficient $\sigma_{i,T}(n)$:

$$A_j(n) = N_j \alpha_j \sigma_{j,T}(n), \qquad A_k(n) = S_k \beta_k \sigma_{k,T}(n), \tag{11}$$

where:

$$\sigma_{j,T}(n) = (N_j - y_j(n))/N_j, \qquad \sigma_{k,T}(n) = (S_k + y_k(n))/S_k. \tag{12}$$

In the case of Erlang streams, the value of the parameter $\sigma_{i,T}(n)$ does not depend on the state of the system and is equal to one. The adopted manner of the presentation of the traffic dependence on the occupancy state of the system allows us to generalize the state-dependent systems (both with state-dependent service process and state-dependant arrival process) and to express all types of dependencies as a product of the corresponding state-passage coefficients.

Having the traffic values: $A_i$ (Eq. (1)), $A_j(n)$ (Eq. (11)) and $A_k(n)$ (Eq. (11)), Generalized Kaufman-Roberts Recursion can be rewritten in the form that includes characteristics of Erlang, Engset and Pascal traffic streams, namely:

$$\begin{aligned} n\,[P_n]_V = &\sum\nolimits_{i=1}^{M_1} A_i t_i\,[P_{n-t_i}]_V + \\ &+ \sum\nolimits_{j=1}^{M_2} N_j \alpha_j \sigma_{j,T}(n-t_j) t_j\,\left[P_{n-t_j}\right]_V + \sum\nolimits_{k=1}^{M_3} S_k \beta_k \sigma_{k,T}(n-t_k) t_k\,[P_{n-t_k}]_V. \end{aligned} \tag{13}$$

The state probabilities, obtained on the basis of Eq. (13), constitute the input data for the next iteration, where the parameters $y_j(n)$, $y_k(n)$ and subsequently $\sigma_{j,T}(n)$, $\sigma_{k,T}(n)$ are designated. The iterative process ends when the assumed accuracy is obtained.

To sum up, the algorithm of determination of the occupancy distribution in systems with multi-service Erlang, Engset and Pascal streams may be written as follows:

| No. | $V$ | $M$ | Traffic structure | | | |
|---|---|---|---|---|---|---|
| | | | $t_1$ | $t_2$ | $t_3$ | $t_4$ |
| 1 | 10 | 2 | 1 | 2 | — | — |
| 2 | 12 | 3 | 1 | 2 | 3 | — |
| 3 | 30 | 3 | 1 | 2 | 6 | — |
| 4 | 64 | 4 | 1 | 2 | 4 | 10 |
| 5 | 150 | 3 | 1 | 2 | 10 | — |
| 6 | 150 | 4 | 1 | 2 | 10 | 20 |

Table 1. Considered full-availability groups

1. Determination of initial values of $y_j^{(0)}(n)$, $y_k^{(0)}(n)$, $E_i^{(0)}$, $E_j^{(0)}$ and $E_k^{(0)}$:
   $\forall_{1\le j\le M_2}\forall_{0\le n\le V}\ y_j^{(0)}(n)=0,\ \forall_{1\le k\le M_3}\forall_{0\le n\le V}\ y_k^{(0)}(n)=0,$
   $E_i^{(0)}=0,\ E_j^{(0)}=0,\ E_k^{(0)}=0.$
2. Determination of state probabilities $[P_n^{(l)}]_V$ (Eq. (13)) in the iteration No. $l$.
3. Determination of blocking probabilities $E_i^{(l)}$, $E_j^{(l)}$ and $E_k^{(l)}$ for calls of particular traffic classes (Eq. (3)).
4. Calculation of reverse transition rates $y_j^{(l)}(n)$ and $y_k^{(l)}(n)$ on the basis of Eq. (4), and subsequently the state-passage coefficients $\sigma_{j,T}^{(l)}(n)$ and $\sigma_{k,T}^{(l)}(n)$ on the basis of Eq. (12).
5. Repeat Steps No. 2, 3 and 4 until the assumed accuracy of the iterative process is obtained:

$$\left|\frac{E_i^{(l-1)}-E_i^{(l)}}{E_i^{(l)}}\right|\le\xi,\quad \left|\frac{E_j^{(l-1)}-E_j^{(l)}}{E_j^{(l)}}\right|\le\xi,\quad \left|\frac{E_k^{(l-1)}-E_k^{(l)}}{E_k^{(l)}}\right|\le\xi. \tag{14}$$

The universal nature of Equation (13) should be particularly stressed. Depending on a type of the offered traffic, the equation can determine the occupancy distribution in systems with just one type of traffic, for example only Pascal traffic, where $M_1 = 0$, $M_2 = 0$ i $M_3 \neq 0$), or in systems with the mixture of traffic from different types of sources as in, for instance, Engset and Pascal types of traffic, when $M_1 = 0$, $M_2 \neq 0$ i $M_3 \neq 0$)

## 4. NUMERICAL EXAMPLES

In order to evaluate the accuracy of the proposed Multiple Iteration Method, the results of analytical calculations are compared to data obtained on the basis of other known analytical methods, i.e. to the exact Iversen convolution method and to the Single Iteration Method. Calculations are carried out for six different FAGs characterized in Table 4. by specifying the capacity $V$ of the group and the number of bandwidth units demanded for calls of particular traffic classes. The groups are offered various number of BPP traffic classes in the following

| | class 1 | | | class 2 | | |
|---|---|---|---|---|---|---|
| $a$ | MIM | SIM | Iversen | MIM | SIM | Iversen |
| 0.4 | 0.000338 | 0.000325 | 0.000338 | 0.000746 | 0.000717 | 0.000746 |
| 0.5 | 0.001636 | 0.001552 | 0.001636 | 0.003535 | 0.003356 | 0.003535 |
| 0.6 | 0.005166 | 0.004852 | 0.005166 | 0.010962 | 0.010309 | 0.010962 |
| 0.7 | 0.012053 | 0.011248 | 0.012053 | 0.025162 | 0.023523 | 0.025162 |
| 0.8 | 0.022673 | 0.021088 | 0.022673 | 0.046614 | 0.043458 | 0.046614 |
| 0.9 | 0.036617 | 0.034023 | 0.036617 | 0.074200 | 0.069146 | 0.074200 |
| 1.0 | 0.053094 | 0.049357 | 0.053094 | 0.106101 | 0.098983 | 0.106101 |
| 1.1 | 0.071288 | 0.066369 | 0.071288 | 0.140550 | 0.131393 | 0.140550 |
| 1.2 | 0.090522 | 0.084457 | 0.090522 | 0.176140 | 0.165117 | 0.176140 |
| | class 3 | | | class 4 | | |
| 0.4 | 0.001821 | 0.001753 | 0.001821 | 0.008559 | 0.008277 | 0.008559 |
| 0.5 | 0.008266 | 0.007863 | 0.008266 | 0.033316 | 0.031912 | 0.033316 |
| 0.6 | 0.024677 | 0.023273 | 0.024677 | 0.087298 | 0.083104 | 0.087298 |
| 0.7 | 0.054735 | 0.051364 | 0.054735 | 0.172700 | 0.164028 | 0.172700 |
| 0.8 | 0.098243 | 0.092033 | 0.098243 | 0.279789 | 0.266013 | 0.279789 |
| 0.9 | 0.151808 | 0.142304 | 0.151808 | 0.393953 | 0.375796 | 0.393953 |
| 1.0 | 0.211046 | 0.198267 | 0.211046 | 0.503029 | 0.482062 | 0.503029 |
| 1.1 | 0.272141 | 0.256472 | 0.272141 | 0.599944 | 0.577927 | 0.599944 |
| 1.2 | 0.332356 | 0.314393 | 0.332356 | 0.682006 | 0.660450 | 0.682006 |

Table 2: Blocking probability in the system No. 4 with Erlang (class 1), Pascal (class 2) and Engset (class 3 and 4) traffic streams, the number of Pascal and Engset traffic sources is equal to 30; $\xi = 0.000001$

proportions: $A_1 t_1 : A_2 t_2 : \ldots : A_M t_M = 1 : 1 : \ldots : 1$. The blocking probability results in the exemplary FAG No. 4 are presented in Tab. 4. in relation to the value of total traffic offered to a single BBU:

$$a = \frac{\sum_{i=1}^{M_1} A_i + \sum_{j=1}^{M_2} N_j \alpha_j + \sum_{k=1}^{M_3} S_k \beta_k}{V}.$$

In order to determine the influence of the number of iterations on the accuracy of the proposed method, Tab. 4. contains the results of relative errors of blocking probabilities obtained in the FAG No. 2. As the reference method, in determination of the relative errors (RE) for class $i$ blocking probabilities, the exact Iversen convolution method is used, i.e.:

$$RE_i = \left| \frac{E_i^{(l)} - E_i^{Iversen}}{E_i^{Iversen}} \right|, \tag{15}$$

where $l$ is the iteration number. The results presented in Tab. 4. indicate that the proposed iterative method converges very quickly.

All the presented results (Tabs 4. and 4.) show the high accuracy of the proposed MIM. The obtained results are equal to the results determined on the basis of the convolution Iversen algorithm. A lot of simulation experiments carried out by the authors so far indicate that

| $a$ | The number of iteration | | | | | |
|---|---|---|---|---|---|---|
| | SIM | 3 | 5 | 7 | 10 | 15 |
| 0.1 | 1.07E-03 | 5.34E-04 | 0 | 0 | 0 | 0 |
| 0.2 | 2.69E-03 | 4.53E-04 | 0 | 0 | 0 | 0 |
| 0.3 | 1.54E-02 | 1.74E-03 | 1.22E-04 | 9.72E-06 | 0 | 0 |
| 0.4 | 3.06E-02 | 3.93E-03 | 3.88E-04 | 2.69E-05 | 0 | 0 |
| 0.5 | 4.59E-02 | 7.09E-03 | 8.95E-04 | 7.57E-05 | 0 | 0 |
| 0.6 | 6.04E-02 | 1.12E-02 | 1.70E-03 | 1.66E-04 | 0 | 0 |
| 0.7 | 7.37E-02 | 1.61E-02 | 2.86E-03 | 3.13E-04 | 3.01E-07 | 0 |
| 0.8 | 8.58E-02 | 2.18E-02 | 4.39E-03 | 5.28E-04 | 2.15E-07 | 0 |
| 0.9 | 9.67E-02 | 2.80E-02 | 6.34E-03 | 8.22E-04 | 3.29E-07 | 0 |
| 1.0 | 1.07E-01 | 3.48E-02 | 8.70E-03 | 1.20E-03 | 3.96E-07 | 0 |
| 1.1 | 1.16E-01 | 4.21E-02 | 1.15E-02 | 1.68E-03 | 5.49E-07 | 0 |
| 1.2 | 1.24E-01 | 4.97E-02 | 1.47E-02 | 2.25E-03 | 7.51E-07 | 0 |
| 1.3 | 1.32E-01 | 5.76E-02 | 1.84E-02 | 2.92E-03 | 9.03E-07 | 0 |
| 1.4 | 1.39E-01 | 6.58E-02 | 2.25E-02 | 3.69E-03 | 1.02E-06 | 0 |
| 1.5 | 1.45E-01 | 7.42E-02 | 2.70E-02 | 4.56E-03 | 1.19E-06 | 0 |

Table 3: Relative errors of blocking probabilities results, obtained on the basis of MIM for class 1 calls, in relation to the number of iterations; System No. 2 with three Engset traffic streams; the number of traffic sources of particular traffic classes is equal to 6; the calculations performed for the blocking probability results determined with the precision set to 8 digits

similar accuracy can be obtained for greater group capacity and greater number of traffic classes serviced by the FAG with a BPP traffic streams. It can be notice that the advantage of MIM is visible in case of systems with low number of traffic sources. For the ratio of number of traffic sources and the group capacity higher than 5, both methods: SIM and MIM offer the same accuracy.

## 5. CONCLUSIONS

In this paper the new approximate method, the so-called Multiple Iteration Methods (MIM), of blocking probability calculations in the systems with Engset-Erlang-Pascal multi-rate traffic streams has been proposed. The method is based on a simple modification of the generalized Kaufman-Roberts recursion. The accuracy of the proposed method has been compared with the Iversen convolution algorithm. This research has confirmed that the MIM can assure the same accuracy as the Iversen algorithm in the case of the full-availability group, i.e. in the case of the system with state-independent service process. However, the main advantage of the proposed method is the possibility of blocking probability calculation in the systems with state-dependent service process (i.e. the model limited-availability group, the model of a single link with bandwidth reservation). Calculations made according to the proposed formulae are not complicated and are easily programmable.

## References


[1] V.B. Iversen, editor. *Teletraffic Engineering Handbook*. ITU-D, Study Group 2, Question 16/2, Geneva, December 2003.

[2] Y. Kogan and M. Shenfild. Asymptotic solution of generalized multiclass Engset model. In J. Labetoulle and J.W. Roberts, editors, *Proceedings of 14th International Teletraffic Congress*, volume 1b, pages 1239–1249, Antibes Juan-les-Pins, France, 1994. Elsevier.

[3] G.L. Choudhury, K.K. Leung, and W. Whitt. An inversion algorithm to compute blocking probabilities in loss networks with state-dependent rates. *IEEE/ACM Trans. Networking*, 3:585–601, 1995.

[4] S.A. Berezner and A.E. Krzesinski. An efficient stable recursion to compute multiservice blocking probabilities. *Journal of Performance Evaluation*, 43(2–3):151–164, 2001.

[5] K.W. Ross. *Multiservice Loss Models for Broadband Telecommunication Network*. Springer, London, 1995.

[6] A.A. Nilson and M.J. Perry. Multi-rate blocking probabilities: numerically stable computation. In V. Ramaswami and P.E. Wirth, editors, *Proceedings of 15th International Teletraffic Congress*, pages 1359–1368, Washington D.C., USA, 1997. Elsevier.

[7] G.M. Stamatelos and J.F. Hayes. Admission-control technics with application to broadband networks. *Computer Communication*, 17(9):663–673, September 1994.

[8] J. Conradt and A. Buchheister. Considerations on loss probability of multi-slot connections. In *Proceedings of 11th International Teletraffic Congress*, pages 4.4B–2.1, Kyoto, Japan, 1985.

[9] J.M. Karlsson. Loss performance in trunk groups with different capacity demands. In *Proceedings of 13th International Teletraffic Congress*, volume Discussion Circles, pages 201–212, Copenhagen, Denmark, 1991.

[10] V.B. Iversen. The exact evaluation of multi-service loss systems with access control. In *Seventh Nordic Teletraffic Seminar (NTS-7)*, pages 56–61, Lund,Sweden, August 1987.

[11] M.E. Beshai and D.R. Manfield. Multichannel services performance of switching networks. In *Proceedings of 12th International Teletraffic Congress*, pages 857–864, Torino, Italy, 1988. Elsevier.

[12] M. Stasiak. Blocking probability in a limited-availability group carrying mixture of different multichannel traffic streams. *Annales des Télécommunications*, 48(1-2):71–76, 1993.

[13] M. Stasiak. An approximate model of a switching network carrying mixture of different multichannel traffic streams. *IEEE Transactions on Communications*, 41(6):836–840, 1993.

[14] L.E.N. Delbrouck. On the steady-state distribution in a service facility carrying mixtures of traffic with different peakedness factors and capacity requirements. *IEEE Transactions on Communications*, 31(11):1209–1211, 1983.

[15] J.S. Kaufman. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 29(10):1474–1481, 1981.

[16] J.W. Roberts. A service system with heterogeneous user requirements — application to multi-service telecommunications systems. In G. Pujolle, editor, *Proceedings of Performance of Data Communications Systems and their Applications*, pages 423–431, Amsterdam, 1981. North Holland.

[17] R. Fortet and C. Grandjean. Congestion in a loss system when some calls want several devices simultaneously. *Electrical Communication*, 39(4):513–526, 1964.

[18] J.W. Roberts. Teletraffic models for the Telcom 1 integrated services network. In *Proceedings of 10th International Teletraffic Congress*, page 1.1.2, Montreal, Canada, 1983.

[19] M. Głąbowski and M. Stasiak. An approximate model of the full-availability group with multi-rate traffic and a finite source population. In P. Buchholtz, R. Lehnert, and M. Pióro, editors, *Proceedings of 3rd Polish-German Teletraffic Symposium*, pages 195–204, Dresden, Germany, September 2004. VDE Verlag.

[20] M. Głąbowski and M. Stasiak. Multi-rate model of the limited-availability group with finite source population. In K. Gong, Z. Niu, P. Fan, and J. Yang, editors, *Proceedings of The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications*, volume 1, pages 366–370, Beijing, August 29 – September 1 2004. IEEE Press.

[21] J.W. Roberts, V. Mocci, and I. Virtamo, editors. *Broadband Network Teletraffic, Final Report of Action COST 242*. Commission of the European Communities, Springer, Berlin, 1996.

[22] J.W. Roberts, editor. *Performance Evaluation and Design of Multiservice Networks, Final Report COST 224*. Commission of the European Communities, Brussels, 1992.

[23] M. Stasiak and M. Głąbowski. A simple approximation of the link model with reservation by a one-dimensional Markov chain. *Journal of Performance Evaluation*, 41(2–3):195–208, July 2000.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 89 - 98

# IM-OM matching packet dispatching scheme for MSM Clos-network switches

JANUSZ KLEBAN SŁAWOMIR WĘCLEWSKI

Chair of Telecommunication and Computer Networks
Faculty of Electronics and Telecommunications
Poznan University of Technology
Ul. Polanka 3, 60-965 Poznań, Poland
*janusz.kleban@et.put.poznan.pl*
*slawomir.weclewski@et.put.poznan.pl*

**Abstract:** Current packet dispatching algorithms for next generation network nodes (switches/routers), in the majority of cases, involve the request-grant-accept handshaking scheme with many iterations and the effect of desynchronization of arbitration pointers. For high-performance switches and routers the Clos switching fabric is very attractive because of its modular architecture and scalability. It is almost impossible to implement the algorithms with multiple phase iterations in the three-stage Clos-network environment with currently available technologies, as the arbitration signals need to pass through the SERDES links several times, and the delay is too long. A great many packet dispatching schemes for three-stage buffered and bufferless Clos switching fabrics were proposed in the literature. Some of them provide high throughput under uniform, others under nonuniform traffic distribution. In this paper the new packet dispatching scheme, called IM-OM Matching (IOM), is proposed and evaluated. We have eliminated the handshaking process and iterations, but it is necessary to use the central arbiter instead. IOM scheme may be implemented in MSM as well as in modified MSM Clos-network switches. We show via simulation that proposed scheme delivers very good performance in terms of throughput, cell delay and input buffers size under different traffic distribution patterns.

**Keywords:** Clos-network, Dispatching Algorithm, Packet Switching, Packet Scheduling

## 1. Introduction

The Internet cannot continue to scale-up to higher data rates without network nodes (switches/routers) with high-speed interfaces and large switching capacity. Current router technology available in the market cannot provide large switching capacity to satisfy future demands. The very fast packet network nodes are still the critical bottleneck. In the architecture of high-performance packet switching nodes the switching function is implemented by using switching fabric boards instead of a shared central bus. The switching fabric transfers input signals to requested outputs through connecting paths and

Scientific work financed from science funding resources in the years 2005-2008 as a research project (grant 3T11D 003 29).

may be built as a single stage switch (e.g. crossbar) or a multi-stage switch, such as the Clos switching fabric. High-speed switching fabrics adopt the use of fixed-length packets called cells. All incoming variable-length packets (e.g. IP packets) are first terminated at ingress line cards, where they are segmented into fixed-size cells, and re-assembled into packets at egress line cards after switching process, before they depart [1]. While cells are being routed in a switching fabric, it is very likely that more than one cell is destined for the same output port or for a physical link inside the multi-stage switching fabric. Cells that have lost contention must be either discarded or buffered. In the case of buffering cells, an arbiter selects only one cell for each output port from among the cells destined for that output port before transmitting the cell. Buffers can be placed at inputs, outputs, inputs and outputs, and/or within the switching fabric. Depending on the buffer placement, respective switches are called input queued (IQ), output queued (OQ), combined input/output queued (CIOQ) and combined input/crosspoint queued (CICQ) [2].

The virtual output queuing (VOQ) is recently widely considered in the literature as a good solution for IQ switches. It removes from them the head-of-line (HOL) blocking problem. HOL blocking causes the idle output to remain idle even if at an idle input there is a cell waiting to be sent to an (idle) output. In VOQ an input buffer in each input port is divided into $N$ parallel queues, each of them storing packets directed to different output ports. When a new cell arrives at the input port, it is stored in the destined queue and waits for transmission through a switching fabric. In this architecture, the memory speed remains compatible with the line rate, but a good matching algorithm between inputs and outputs is needed, so that it can achieve high throughput and low latency.

In VOQ switches internal blocking and output port contention problems are resolved by fast arbitration schemes. An arbitration scheme is essentially a service discipline that arranges the service order among the input cells. In the distributed manner, each output has its own arbiter, operating independently from others. The arbitration scheme decides which items of information should be passed from inputs to arbiters, and – based on that decision – how each arbiter picks one cell from among all input cells destined for the output. Algorithms which can assign the route between input and output modules are usually called packet dispatching schemes. Considerable work has been done on scheduling algorithms for VOQ switches. Most of them achieve 100% throughput under the uniform traffic, but the throughput is usually reduced under the nonuniform traffic [1, 3-13].

Multiple-stage Clos-network switches are a potential solution to overcome the limited scalability of single-stage switches, in terms of number of I/O chip pins and the number of switching elements. In a Clos-network switch packet scheduling is needed as there is a large number of points where contention may occur. The three-stage Clos switching fabric is widely investigated in many papers. It is possible to categorize the three-stage Clos-network switch architecture into two types: bufferless and buffered. The former one has no memory in any stage, and it is also referred to as the space-space-space ($S^3$) Clos-network switch, while the latter one employs shared memory modules in the first and third stages, and it is referred to as the memory-space-memory (MSM) Clos-network

switch. The buffers in the second stage modules cause an out-of-sequence problem, so a re-sequencing function unit in the third stage modules is necessary, but difficult to implement, when the port speed increases. Different dispatching schemes for the three-stage Clos-network switches were proposed in the literature [4-6, 9-13]. The well known dispatching algorithms for the buffered Clos-network switches were proposed in [4-6]. The basic idea of these algorithms is to use the effect of desynchronization of arbitration pointers in the Clos-network switch and common request-grant-accept handshaking scheme. It is very difficult to implement these schemes in the real environment because of time constraints.

In this paper the Input Module – Output Module (IM-OM) matching packet dispatching scheme (IOM) is proposed. This algorithm gives better performance results than other dispatching schemes proposed for the MSM Clos switching fabric, is less demanding in terms of hardware (in comparison with previously proposed schemes) and can achieve 100% throughput for both the uniform and the nonuniform traffic distribution patterns.

The remainder of this paper is organized as follows. Section 2 introduces some background knowledge concerning the MSM Clos switching fabric that we refer to throughout this paper. Section 3 presents the IM-OM matching packet dispatching scheme. Section 4 is devoted to performance evaluation of the proposed IOM algorithm. We conclude this paper in section 5.

## 2. MSM Clos switching fabric

Clos-networks are well known and widely analyzed in the literature [14]. The three-stage Clos-network architecture is denoted by *C(m, n, k)*, where parameters *m, n*, and *k* entirely determine the structure of the network. There are *k* input switches of capacity $n \times m$ in the first stage, *m* switches of capacity $k \times k$ in the second stage, and *k* output switches of capacity $m \times n$ in the third stage. The capacity of this switching system is $N \times N$, where $N = nk$. The three-stage Clos switching fabric is strictly nonblocking if $m \geq 2n-1$ and rearrangeable nonblocking if $m \geq n$.

We define the MSM Clos switching fabric based on the terminology used in [4] (see Fig. 1 and Tab. 1).

In the MSM Clos switching fabric architecture the first stage consists of *k* IMs, and each of them has an $n \times m$ dimension and *nk* VOQs to eliminate Head-Of-Line blocking. The second stage consists of *m* bufferless CMs, and each of them has a $k \times k$ dimension. The third stage consists of *k* OMs of capacity $m \times n$, where each *OP(j, h)* has an output buffer. Each output buffer can receive at most *m* cells from *m* CMs, so a memory speedup is required here.

Generally speaking, in the MSM Clos switching fabric architecture each *VOQ(i, j, h)* located in *IM(i)* stores cells going from *IM(i)* to the *OP(j, h)* at *OM(j)*. In one cell time slot VOQ can receive at most *n* cells from *n* input ports and send one cell to any CMs. A memory speedup of *n* is required here, because the rate of memory work has to be *n* times higher than the line rate. Each *IM(i)* has *m* output links connected to each *CM(r)*,

respectively. A *CM(r)* has *k* output links *LC(r, j),* which are connected to each *OM(j)*, respectively.

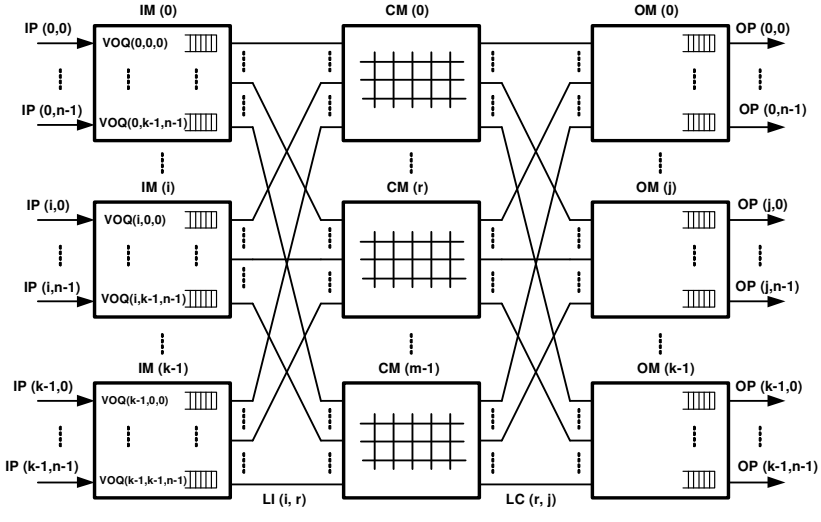



Fig. 1. The MSM Clos switching fabric architecture.

| | |
|---|---|
| IM | Input module at the first stage |
| CM | Central module at the second stage |
| OM | Output module at the third stage |
| *i* | IM number, where $0 \le i \le k-1$ |
| *j* | OM number, where $0 \le j \le k-1$ |
| *h* | Input/output port number in each IM/OM, where $0 \le h \le n-1$ |
| *r* | CM number, where $0 \le r \le m-1$ |
| *IM (i)* | The *(i+1)*th input module |
| *CM (r)* | The *(r+1)*th central module |
| *OM (j)* | The *(j+1)*th output module |
| *IP (i, h)* | The *(h+1)*th input port at IM(i) |
| *OP (j, h)* | The *(h+1)*th output port at OM(j) |
| *LI (i, r)* | Output link at *IM(i)* that is connected to *CM(r)* |
| *LC (r, j)* | Output link at *CM(r)* that is connected to *OM(j)* |
| *VOQ (i, j, h)* | Virtual output queue at IM*(i)* that stores cells from *IM(i)* to *OP(j, h)* |

Tab. 1. A notation for the MSM Clos switching fabric.

We propose to use Virtual Output Module Queues (VOMQs), instead of VOQs. In this case, an input buffer in each IM is divided into *k* parallel queues, each of them storing cells destined to different OMs. It is possible to arrange buffers in such way because OMs are nonblocking. Memory speedup of *n* is necessary here. There are less queues in each IMs but they are longer than VOQs. Each *VOMQ(i, j)* stores cells going from *IM(i)* to the *OM(j).*

The modified MSM Clos switching fabric, proposed by us in [12], is shown in Figure 2.

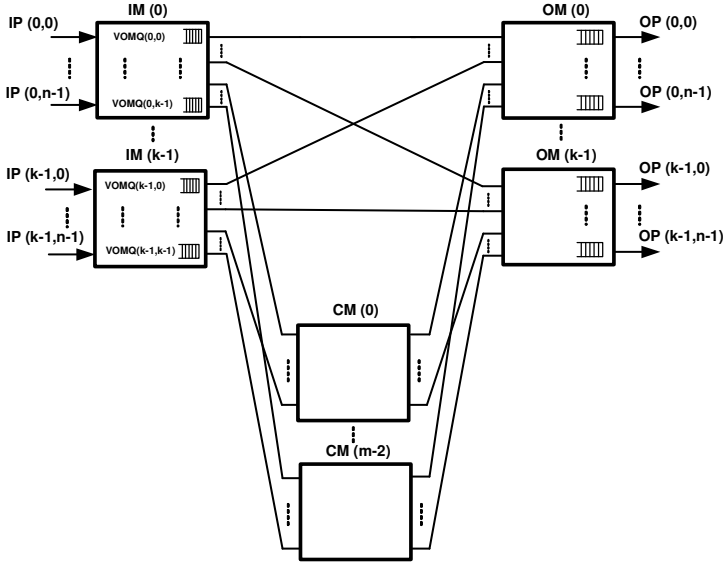



Fig. 2. The modified MSM Clos switching fabric.

We have proposed to change the architecture of MSM Clos switching fabric so as to give the possibility of rapid unload of VOMQs. The main idea of the modification is connecting bufferless CMs to the two-stage buffered switching fabric. In this way an expansion in IMs and OMs is used, however very simple and effective packet dispatching scheme may be implemented. The maximum number of connected CMs is equal to *m-2*, but it is possible to use less CMs. In practice, the number of CMs significantly influences the performance of the switching fabric; the number of CMs depends on the traffic distribution pattern to be set up.

## 3. IM-OM matching packet dispatching scheme

The IOM packet dispatching scheme may be implemented in the MSM Clos-network switches as well as in the modified architecture proposed by us. In [12] we have presented packet dispatching scheme, called Static Dispatching with Rapid Unload of Buffers (SDRUB), which is a switching fabric-oriented algorithm and requires the modified MSM Clos switching fabric. The scheme proposed in this paper sends to the central arbiter different kind of information than SDRUB scheme and requires the arbiter with completely different functionality.

The IOM packet dispatching scheme makes a matching between each IM and OM taking into account the number of cells waiting in VOMQs. Each VOMQ has its own counter *PV(i, j),* which shows the number of cells destined to *OM(j)*. The value of

*PV(i, j)* is increased by 1 when a new cell is written to a memory and decreased by 1 when the cell is sent out to *OM(j)*. IOM algorithm uses a central arbiter to indicate the matched pairs of *IM(i)-OM(j)*. After a matching phase, in the next time slot *IM(i)* is allowed to send cells to selected *OM(j)*.

In detail, the IOM algorithm works as follows:

- Step 1 (each IM): Sort the values of *PV(i, j)* in descending order. On the basis of sorted values send the OMs identifiers to the central arbiter. The identifier of *OM(j),* to which *VOMQ(i, j)* stores the most number of cells send as a first one, and the identifier of *OM(s),* to which *VOMQ(i, s)* stores the least number of cells send as the last one.
- Step 2 (central arbiter): The central arbiter analyzes the request received from *IM(i)* and checks if it is possible to match this IM with *OM(j)*, which identifier was sent as a first one on the list in the request. If the matching is not possible, because the *OM(j)* is matched with other IM, the arbiter selects the next OM on the list. The round-robin arbitration is employed for selection of *IM(i*), which request is analyzed as a first one.
- Step 3 (central arbiter): The central arbiter sends to each IM confirmation with the identifier of *OM(t)*, to which the IM is allowed to send cells.
- Step 4 (each IM): Match all output links *LI(i, r)* with cells from *VOMQ(i, t).* If there is less than n cells to be send to *OM(t)*, some output links remain unmatched.
- Step 5 (each IM): Decrease the value of *PV(i, t)* by the number of cells, which will be send to *OM(t)*.
- Step 6 (each IM): In the next time slot send the cells from the matched *VOMQ(i, t)* to the *OM(t)* selected by the central arbiter.

In the modified MSM Clos switching fabric it is always possible to sent one cell form IM to OM without an arbitration process, because there are direct connections from IMs to OMs. This is the only difference in implementation of IOM scheme in the MSM Clos switching fabric and in the modified architecture.

## 4. Simulation experiments

The Bernoulli packet arrival model is considered in the paper. Cells arrive at each input in slot-by-slot manner. We consider several traffic distribution models which determine the probability that a cell which arrives at an input will be directed to a certain output. The considered traffic models are:

*Uniform traffic* – this type of traffic is the most commonly used traffic profile. In the uniformly distributed traffic probability $p_{ij}$ that a packet from input *i* will be directed to output *j* is uniformly distributed through all outputs, i.e.:

$$p_{ij} = p / N \quad \forall i, j \tag{1}$$

*Nonuniform traffic* – in this traffic model some outputs have a higher probability of being selected, and respective probability $p_{ij}$ was calculated according to the following equation:

$$p_{ij} = \begin{cases} \frac{p}{2} & for \quad i = j \\ \frac{p}{2(N-1)} & for \quad i \neq j \end{cases} \tag{2}$$

*Diagonal traffic* – is very similar to the nonuniform traffic but packets are directed to one of two outputs, and respective probability $p_{ij}$ was calculated according to the following equation:

$$p_{ij} = \begin{cases} \frac{2}{3} p & for \quad i = j \\ \frac{p}{3} & for \quad j = (i+1) \bmod N \\ 0 & otherwise \end{cases} \tag{3}$$

*Chang's traffic* – this model is defined as:

$$p_{ij} = \begin{cases} 0 & for \quad i = j \\ \frac{p}{N-1} & otherwise \end{cases} \tag{4}$$

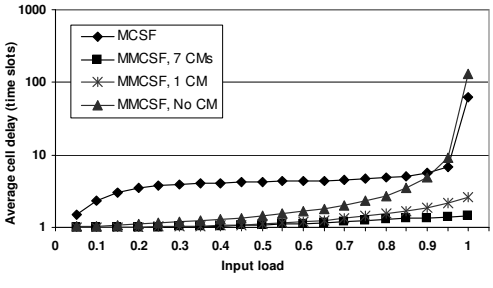



Fig. 3. Average cell delay for IOM algorithm, uniform traffic.

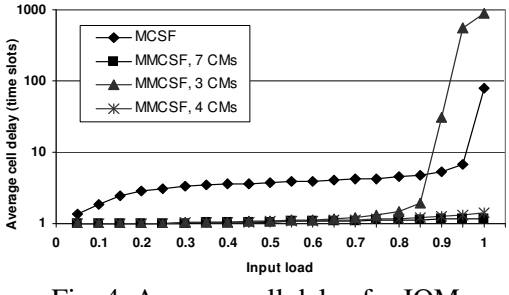



Fig. 4. Average cell delay for IOM algorithm, nonuniform traffic

The experiments have been carried out for the MSM Clos switching fabric (denoted as MCSF) and the modified MSM Clos switching fabric (denoted as MMCSF) of size 64 × 64 - *C(8, 8, 8),* and for a wide range of traffic load per input port: from $p = 0.05$ to $p = 1$, with the step 0.05. The 95% confidence intervals that have been calculated after *t*-student distribution for ten series with 50000 cycles (after the starting phase comprising 15 000 cycles, which enables to reach the stable state of the switching fabric) are at least one order lower than the mean value of the simulation results, therefore they are not shown in the figures. We have evaluated two performance measures: average cell delay in time slots and maximum VOMQs size (we have investigated the worst case). The results of the

simulation are shown in the charts (Fig. 3-10). Fig. 3-6 show the average cell delay in time slots obtained for the uniform, nonuniform, diagonal, and Chang's traffic patterns, whereas Fig. 7-10 show the maximum VOMQ size in number of cells.

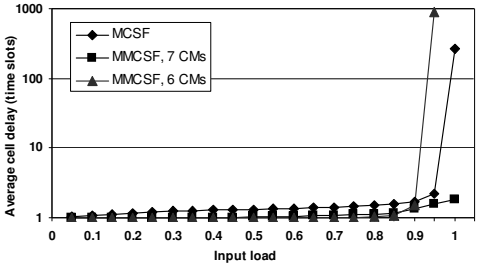



Fig. 5. Average cell delay for IOM algorithm, diagonal traffic

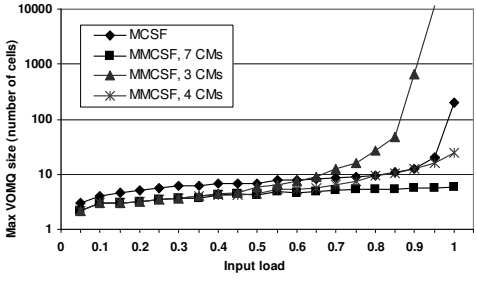



Fig. 8. The maximum VOMQ size, nonuniform traffic.

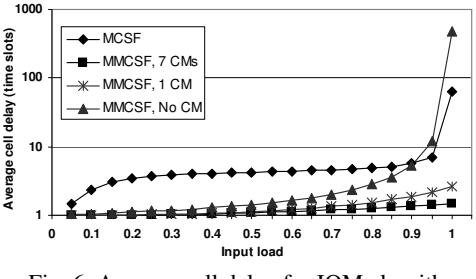



Fig. 6. Average cell delay for IOM algorithm, Chang's traffic.

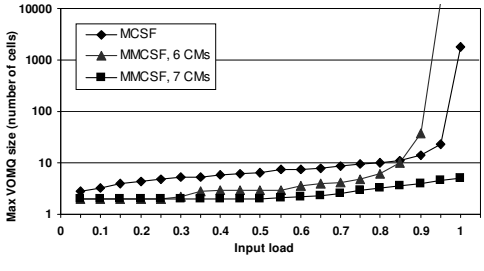



Fig. 9. The maximum VOMQ size, diagonal traffic.

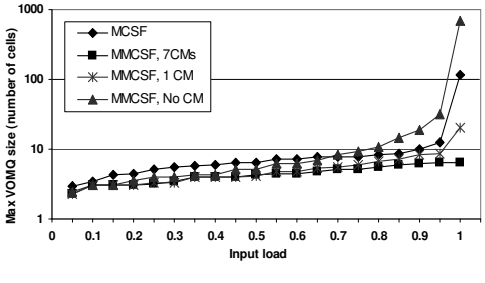



Fig. 10. The maximum VOMQ size, Chang's traffic.

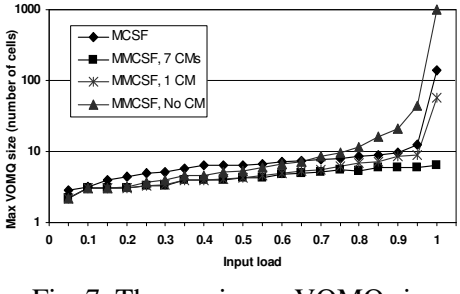



Fig. 7. The maximum VOMQ size, uniform traffic.

We can see that MSM Clos switching fabric with IOM scheme has 100% throughput for all kind of investigated traffic patterns. The average cell delay is less than 10 for wide range of input load regardless of a traffic distribution pattern. It is very interesting results especially for the nonuniform and the diagonal traffic patterns. Both traffic patterns are very demanding and many packet dispatching schemes, proposed in the literature, cannot provide the 100% throughput for the investigated switching fabric.

In all cases the modified MSM Clos switching fabric provides better performance than the basic MSM structure. For the uniform traffic and Chang's traffic a single CM in modified structure significantly improves the switching fabric performance in terms of the average cell delay and the maximum VOMQ size (Fig. 6 and 10). The results are better than for the MSM Clos switching fabric even though this switching fabric has less crosspoints than the basic structure. The investigated MSM Clos switching fabric - *C(8, 8, 8)*- has 1536 crosspoints and the modified architecture with one CM – 1216. To manage the nonuniform traffic effectively in the modified Clos switching fabric, it is necessary to implement at least 4 (*n/2*) CMs (Fig. 4, 8). For such number of CMs the switching fabric has 100% throughput. Any smaller number of CMs reduces the throughput of this switching fabric. Fig. 6 shows that the modified Clos switching fabric with the IOM algorithm has 100% throughput under the diagonal traffic only when the maximum number of CMs is used.

For all kind of investigated traffic patterns the maximum size of VOMQ is close to or less than 10 cells for the modified MSM Clos switching fabric with the IOM scheme. The size of VOMQ in the MSM Clos switching network depends on the traffic distribution pattern. For the uniform and Chang's traffic the maximum size of VOMQ is less than 140 cells, for the nonuniform traffic is less than 200 cells, and for the diagonal traffic the maximum size of VOMQ is less than 2000.

## 5. Conclusions

We have proposed the IM-OM matching packet dispatching scheme, which may be implemented both in the MSM Clos switching fabric and in the modified MSM Clos switching fabric. This scheme uses the central arbiter to match IMs with OMs. The arbiter performs relatively simple function thanks to sorted list of OM's identifiers, which is sent by each IM. Simulation experiments have shown that the proposed scheme is very promising and gives very good results for the uniform and nonuniform traffic patterns. It can managed all investigated traffic patterns very effectively, which is desirable feature for network nodes in the next generation IP networks. The IOM algorithm seems to be implementable within the current technology in the real environment. A hardware implementation of the central arbiter required by the IM-OM matching packet dispatching scheme will be the subject of further research.

# References

[1] H. J. Chao, C. H. Lam, and E. Oki: "Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers", Willey, New York, 2001.

[2] K. Yoshigoe and K.J. Christensen: "An evolution to crossbar switches with virtual ouptut queuing and buffered cross points"', *IEEE Network*, vol. 17, no. 5, 2003, pp. 48-56.

[3] E. Oki, R. Rojas-Cessa, and H. J. Chao: “A pipeline-based approach for maximal-sized matching scheduling in input-buffered switches”, *IEEE Communications Letters*, vol. 5, no. 6, 2001, pp. 263-265.

[4] E. Oki, Z. Jing, R. Rojas-Cessa, and H. J. Chao: “Concurrent Round-Robin-Based Dispatching Schemes for Clos-Network Switches”, *IEEE/ACM Trans. on Networking*, vol. 10, no.6, 2002, pp. 830-844.

[5] R. Rojas-Cessa, and H. J. Chao: “Maximum Weight Matching Dispatching Scheme in Buffered Clos-Network Packet Switches”, in *Proc. IEEE International Conference on Communications 2004 - ICC-2004*, Paris, France, 2004, pp. 830-844.

[6] K. Pun, M. Hamdi: “Dispatching schemes for Clos-network switches”, *Computer Networks* no. 44, 2004, pp.667-679.

[7] Y. Jiang, M. Hamdi: “A fully desynchronized round-robin matching scheduler for a VOQ packet switch architecture”, in *Proc. IEEE High Performance Switching and Routing 2001 – HPSR 2001,* May 2001, pp. 407–411.

[8] Hui, J.Y. and E. Arthurs, “A Broadband Packet Switch for Integrated Transport”, *IEEE J. Sel. Areas Commun.*, vol. 5, no. 8, Oct. 1987, pp. 1264-1273.

[9] Chuan-Bi Lin and R. Rojas-Cessa: “Frame Occupancy-Based Dispatching Schemes for Buffered Three-stage Clos-Network switches”, in *Proc. 13th IEEE International Conference on Networks 2005*.

[10] R. Rojas-Cessa, and Chuan-Bi Lin: “Scalable Two-stage Clos-Network Switch and Module-First Matching”, in *Proc. High Performance Switching and Routing 2006 – HPSR 2006*, pp. 303-308.

[11] J. Kleban, A. Wieczorek: “CRRD-OG: A packet Dispatching Algorithm with Open Grants for Three-Stage Buffered Clos-Network Switches”, in *Proc. High Performance Switching and Routing 2006 – HPSR 2006*, pp. 315-320.

[12] J. Kleban, M. Sobieraj, S. Węclewski: "The Modified MSM Clos Switching Fabric with Efficient Packet Dispatching Scheme", in *Proc. IEEE High Performance Switching and Routing 2007 – HPSR 2007*, New York, May 30 to June 1, 2007.

[13] J. Kleban, H. Santos: "Packet Dispatching Algorithms with the Static Connection Patterns Scheme for Three-Stage Buffered Clos-Network Switches", in *Proc. IEEE International Conference on Communications 2007 - ICC-2007*, 24-28 June 2007 Glasgow, Scotland.

[14] C. Clos: “A Study of Non-Blocking Switching Networks”, *Bell Sys. Tech. Jour.*, 1953, pp. 406-424.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 99–110

# Uplink and Downlink Blocking Probability Calculation for Cellular Systems with WCDMA Radio Interface and Finite Source Population

MACIEJ STASIAK, ARKADIUSZ WIŚNIEWSKI AND PIOTR ZWIERZYKOWSKI [a]

[a]Chair of Communications and Computer Networks,
Poznań University of Technology, ul. Polanka 3, Poznań 60965, Poland,
e-mail: {stasiak,pzwierz}@et.put.poznan.pl

**Abstract:** The admission control in wireless networks with the WCDMA radio interface admits or blocks new calls depending on a current load situation both in the access cell and in neighboring cells. A new call is rejected if the predicted load exceeds particular thresholds set by the radio network planning. This article presents a new blocking probability calculation method in cellular systems with the WCDMA radio interface for the uplink and the downlink directions. The model considers a finite and an infinite source population of users. In the model, we use the load factor to estimate whether a new call can be admitted or blocked. In the proposed calculation method, the corresponding value of the load factor in the neighboring cell is based on the Okumura-Hata propagation model. The results of the analytical calculations were compared with the results of the simulation experiments, which confirmed the accuracy of the proposed method. The proposed scheme could be applicable for a cost-effective radio resource management in 3G mobile networks and can be easily applied to network capacity calculations.

**Keywords:** UMTS, WCDMA, Blocking Probability

## 1. Introduction

Universal Mobile Telecommunications Systems (UMTS) using the WCDMA radio interface is one of the standards proposed for the third generation cellular technologies (3G). According to the ITU recommendations, 3G systems should include services with circuit switching and packet switching, transmit data with a speed of up to 14 Mbit/s, and ensure access to multimedia services [23]. Due to the possibility of resource allocation for different traffic classes, capacity calculation of the WCDMA radio interface is much more complex than in the case of GSM systems. Moreover, all users serviced by a given cell make use of the same frequency channel and a differentiation of the transmitted signals is possible, only and exclusively, by an application of orthogonal codes [2]. However, due to the multipath propagation occurring in a radio channel, not all transmitted signals are orthogonal with respect to one another and, consequently, are received by users of the system as an interference adversely affecting the capacity of the system. Additionally, the increase in interference is

caused by users serviced by other cells of the system, who make use of the same frequency channel, as well as by the users making use of the adjacent radio channels. To ensure appropriate level of service in UMTS it is thus necessary to limit the interference by decreasing the number of active users or the allocated resources employed to service them.

Several papers have been devoted to traffic modelling in cellular systems with the WCDMA radio interface. In [22], the number of ongoing calls in a cell was modelled as a Poisson random variable and the total interference as a compound Poisson sum, assuming that no call would be blocked. However, such an assumption cannot be fulfilled in a real network. In [18] the total blocking probability in an access cell with infinite source population is calculated with the help of the so-called Kaufman-Roberts recursion [3, 8, 9, 10, 15], while other cell interferences are characterized by a random variable with the lognormal distribution.

The proposed analytical method is the extension of the the methodology proposed by the authors in several earlier works such as [5, 20, 21]. In the presented analytical model we assume that the WCDMA radio interface can be modelled by the full-availability group servicing a mixture of multi-rate Erlang and Engset traffic streams. Additionally, we consider the radio interface in the uplink and the downlink directions for the first time, and introduce a propagation model for the estimation of the influence of the load factor on the neighboring cells for the model of the group cells.

The article has been divided into five sections. Section II discusses basic dependencies describing radio interface load for the uplink and downlink direction. Section III presents an analytical model employed in blocking probability calculations. The next section includes a comparison of the results obtained in the calculation with the simulation data for a system comprising seven cells. The final section sums up the discussion.

## 2. WCDMA radio interface

The WCDMA radio interface offers enormous possibilities in obtaining large capacities. It imposes, however, a substantial number of limits as regards the acceptable level of interference in the frequency channel. In every cellular system with spreading spectrum the radio interface capacity is seriously limited due to the occurrence of some types of interference [23], namely: (1) co-channel interference within a cell – from concurrent users of a frequency channel within the area of a given cell; (2) outer co-channel interference – from concurrent users of the frequency channel working within the area of adjacent cells; (3) adjacent channels interference – from the adjacent frequency channels of the same operator or other cellular telecommunication carriers; (4) all possible noise and interference coming from other systems and sources, both broadband and narrowband.

Accurate signal reception is possible only when the relation of energy per bit $E_b$ to noise spectral density $N_0$ is appropriate [6]. A too low value of $E_b/N_0$ will cause the receiver to be unable to decode the received signal, while a too high value of the energy per bit in relation to noise will be perceived as interference for other users of the same radio channel. The relation

$E_b/N_0$ for a user of the class $i$ service can be calculated as follows [6]:

$$\left(\frac{E_b}{N_0}\right)_i = \frac{W}{\nu_i R_i} \frac{P_i}{I_{total} - P_i}, \tag{1}$$

where: $P_i$ – signal power received from a user of the class $i$ connection, $W$ – chip rate of spreading signal, $v_i$ – activity factor of a user of the class $i$ service, $R_i$ – bit rate of a user of the class $i$ service, $I_{total}$ – total received wideband power, including thermal noise power. The mean power of a user of the class $i$ service:

$$P_i = \left(1 + \frac{W}{(\frac{E_b}{N_0})_i R_i \nu_i}\right)^{-1} I_{total} = L_i I_{total}, \tag{2}$$

where $L_i$ is the load factor for a user of the class $i$ connection. Table 1 shows sample values $E_b/N_0$ for different traffic classes and corresponding values of the load factor $L_i$ [20]. Once $L_i$ of a single user has been established, it is possible to obtain the total load for the uplink [6]:

$$\eta_{UL} = \sum\nolimits_{i=1}^{M} L_i n_i, \tag{3}$$

where $M$ is the number of services and $n_i$ is the number of users of the class $i$ service. The above relation is true when we deal with a system that consists of a single cell. In fact, there are many cells in which the generated traffic influences the capacity of the radio interface of other cells. Thus, (3) should be complemented with an element that would take into consideration the interference from other cells. To achieve this, a variable $\bar{\delta}$ is introduced, which is defined as the mean value of the other cell interference over proper cell interference [6]:

$$\eta_{UL} = \left(1 + \bar{\delta}\right) \sum\nolimits_{i=1}^{M} L_i n_i. \tag{4}$$

The bigger the load of a radio link, the higher level of the noise generated. When the load of the uplink direction approaches unity, the corresponding increase in noise tends towards infinity. Therefore, it is assumed that the actual maximum use of the radio interface without lowering the level of the quality of service will amount to about 50 – 80% [12].

Table 1. Examples of $E_b/N_0$, $\nu_i$ and $L_i$ for different service classes [20]

| Class of service ($i$) | Speech | Video call | Data | Data |
|---|---|---|---|---|
| $W$ [Mchipps] | 3.84 | | | |
| $R_i$ [kbps] | 12.2 | 64 | 144 | 384 |
| $\nu_i$ | 0.67 | 1 | 1 | 1 |
| $(E_b/N_0)_i$ [dB] | 4 | 2 | 1.5 | 1 |
| $L_i$ | 0.0053 | 0.0257 | 0.0503 | 0.1118 |

The downlink direction equation is similar to the uplink direction with the addition of the orthogonality factor $\xi_i$ due to the orthogonality provided by the OVSF (*Orthogonal Variable*

*Spreading Factor*) codes. In the WCDMA, OVSF codes are used to separate downlink direction channels transmitted from a single Node B. Thus, the downlink direction load factor can be calculated [6]:

$$\eta_{\mathrm{DL}} = (1 - \xi_i + \bar{\delta}) \sum_{i=1}^{M} L_i n_i. \tag{5}$$

Typically, the orthogonality is between 0.4 and 0.9 in multipath channels.

In the downlink direction we assume that the only load introduced to the cell is equal to load factor $L_i$ generated by a given class of service in the downlink direction.

## 3. Model of the system

Before admitting a new connection in systems with the WCDMA radio interface, admission control needs to check whether the admittance will not sacrifice the quality of the existing connections. The admission control functionality is located in RNC (*Radio Network Controller*) where the load information from the surrounding cells can be obtained. The admission control algorithm estimates the load increase that would be caused in the radio network by the setting up of a new connection [6]. This is done not only in the access cell but also in the adjacent cells in order to take the inter-cell interference effect into account. A new call is rejected if the predicted load exceeds thresholds set by the radio network planning [12].

### 3.1. Basic assumptions

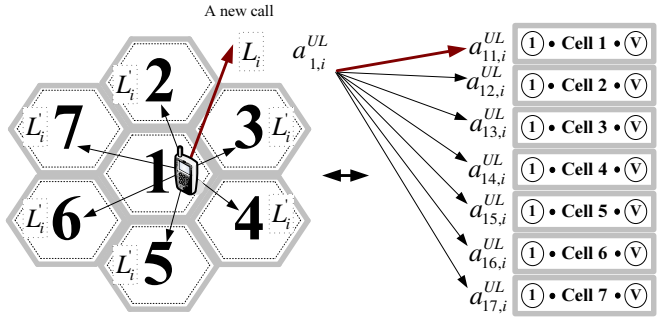
Let us consider a seven-cell system with omnidirectional antennas. The system services a mixture of multi-rate traffic streams in the uplink and the downlink directions. Let us assume that every new class $i$ call is accompanied by load factor $L_i$ (cf.tab. 1). We assume that the calls serviced in the system require resources in two directions (uplink and downlink).

Let us consider a system which services connections in the uplink direction (Fig. 1(a)). In this figure it is assumed that the access cell is, for instance, the cell designated with "1". A new class $i$ call which is offered to the access cell and requires $L_i$ resources is accompanied by load $L_i^{'}$ introduced by the call to radio interfaces of the neighboring cells. We can determine the value of $L_i^{'}$ using one of many propagation models, including semi-deterministic models [1, 17] and models based on ray tracing [17]. In this proposed method we determined the value of $L_i^{'}$ using the Okumura-Hata propagation model [13, 14]. For analytical purposes, the access cell was divided into areas with resolution 0,5 meter. For each point, path loss propagation and signal power received in the neighboring cell were calculated. The average received power was used for calculating the average load factor $\bar{L_i^{'}}$:

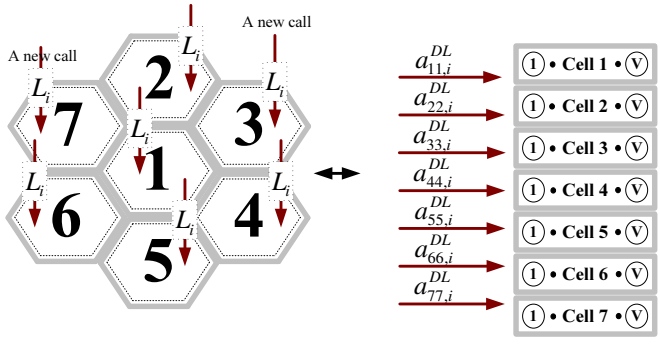
$$L_i^{'} = \bar{L_i^{'}} = \bar{P_i^{'}} / I_{total}, \tag{6}$$

where $\bar{P_i^{'}}$ is the average received signal power in the base station.

Let us consider now the CAC function located in RNC. The operation of the CAC function effects in a rejection of a given call when the level of the usage of the the radio interface in the access cell and the neighboring cells exceeds the assumed threshold. This means that




(a) Cellular system in the uplink direction.




(b) Cellular system in the downlink direction.

Fig. 1: An exemplary cellular system: a seven-cell set system and a set of full-availability groups as the model of wireless cellular network; a) system in the uplink direction; b) system in the downlink direction.

the service processes of a new call occurring in the access cell and those in the neighboring cells are interdependent. These dependencies have been determined on the basis of the fixed-point methodology [5, 11].

Figure 1(b) shows a cellular system servicing a mixture of different mult-rate traffic streams in the downlink direction. In the figure it is assumed that each of the cells is the access cell. A new class $i$ call which is offered to the cell requires $L_i$ resources in the downlink direction in each cell to which it is offered (cf. Sect.2.).

### 3.2. Single cell model

The radio interface in a single UMTS cell in both directions can be treated as a full-availability group (FAG) with multi-rate traffic. The demanded resources in the group for servicing particular classes can be treated as a call demanding an integer number of the so-called BBUs (Basic Bandwidth Units) [16]. Let us assume that the total capacity of the cell is equal to $V$ BBUs. The value of BBU, i.e. $L_{BBU}$, is calculated as the greatest common

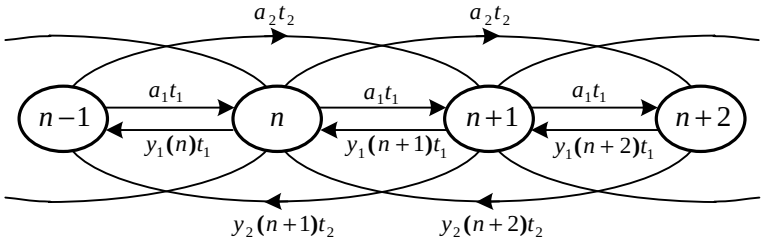



Fig. 2. A fragment of the one-dimensional Markov chain in the FAG.

divisor of all load factors offered to the system:

$$L_{BBU} = GCD(L_1, ..., L_M). \tag{7}$$

The cell is offered $M$ independent classes of Poisson traffic streams having the intensities: $\lambda_1$, $\lambda_2$,...,$\lambda_M$. The class $i$ call requires $L_i$ BBUs to set up a connection. The holding time for calls of particular classes has an exponential distribution with the parameters: $\mu_1$, $\mu_2$,...,$\mu_M$. Thus, the mean traffic offered to the system by the class $i$ traffic stream is equal to: $a_i = \lambda_i/\mu_i$ . The full-availability group (FAG) can be approximated by the Fortet-Grandjean-Kaufman-Roberts recursion [3, 7, 10, 15]:

$$nP_{Er}(n) = \sum\nolimits_{i=1}^{M} a_i t_i P_{Er}\left(n - t_i\right), \tag{8}$$

where $P_{Er}(n)$ is the occupancy distribution in the group with an infinite source population, i.e. the probability of $n$ BBUs being busy ($n \in \langle 0; V \rangle$), and $t_i$ is the number of BBUs required by class $i$ calls, and is defined as follows:

$$t_i = L_i/L_{BBU}. \tag{9}$$

On the basis of (8) we can recurrently calculate the occupancy distribution $P_{Er}(n)$ in the FAG with multi-rate traffic. Having determined all the blocking probabilities $P_{Er}(n)$, we can calculate the blocking probability for class $i$ calls:

$$B_i = \sum\nolimits_{n=V-t_i+1}^{V} P_{Er}(n) \ . \tag{10}$$

Figure 2 presents a one-dimensional Markov chain diagram constructed for a FAG with multi-rate traffic. This figure is a geometrical interpretation of the formula (8). In the figure, symbol $y_i(n)$ denotes the reverse transition rates of a class $i$ service stream outgoing from state $n$. This parameter determines the average number of class $i$ calls serviced in state $n$ [10, 19]. The value of $y_i(n)$, in a given state of the group, forms the basis of the method of occupancy distribution calculation in the FAG with a finite population of traffic sources presented in [4]. The basis for the calculation algorithm, proposed in [4], is the assumption on the equality of reverse transition rates in both infinite and finite population models. According

to this algorithm, in the first step we calculate reverse transition rates $y_i(n)$ in the FAG group with an infinite population of traffic sources. The probabilities $P_{Er}(n)$ in (8) are calculated by the FAG, under the initial assumption that the offered traffic is not dependent on the number of in-service sources, and is equal to:

$$a_i = N_i \alpha_i. \tag{11}$$

Determining reverse transition rates $y_i(n)$ is essential to the proposed method since it is assumed that the number of in-service sources of class $i$ in the aggregated state of $n$ BBUs being busy $n_i(n)$ is approximated by the parameter $y_i(n)$. The determined values of $y_i(n)$ enable us to make the mean value of the offered traffic dependent on the occupancy state of the group in the following way:

$$a_i(n) = (N_i - y_i(n))\alpha_i, \tag{12}$$

and, consequently, to construct the one-dimensional Markov process in the FAG with a finite population of traffic sources.

Eventually, the recurrent equation which determines the occupancy distribution in the FAG with multi-rate traffic and a finite population of sources, can be obtained as follows:

$$nP_{En}(n) = \sum\nolimits_{i=1}^{M} a_i(n - t_i)t_i P_{En}(n - t_i), \tag{13}$$

where $P_{En}(n)$ is the occupancy distribution in the group with a finite source population. The blocking probability $B_i$ in the considered system can be calculated in the following way:

$$B_i = \sum\nolimits_{n=V-t_i+1}^{V} P_{En}(n). \tag{14}$$

### 3.3. Model of the group of cells

Let us consider a group of a seven-cell system with omnidirectional antennas. Each of the cells in the uplink and in the downlink direction can service a mixture of multi-rate traffic generated by an infinite and a finite source population. In the description of the model let us designate the access cell $z$ and let us assume that it is surrounded by six neighboring cells. Additionally, let us assume that the uplink and the downlink direction will be considered separatively. Figures 1(a) and 1(b) show a model of the considered system in which each of the cells in each of the directions is represented by the FAG.

In Figure 1(a) it is assumed that a new call of class $i$ is offered to the radio interface of the cell "1". According to Eqs. (7) and (9), a call requires $t_i$ resources in the access cell and $t_i^{'}$ resources in the neighboring cells. Figure 1(a) shows a traffic distribution scheme for the system under consideration. For Figure 1(a) the following notation is used: $V$ – the cell capacity, $a_{z,i}^{UL}$ – the mean traffic offered to the system by users of the class $i$ in the cell $z$, $a_{zh,i}^{UL}$ – the mean traffic offered in the cell $h$ by user of class $i$ in the cell $z$.

In the proposed model it is assumed that a new call is rejected when the assumed increase in the load, both in the access cell or the neighboring cells, exceeds the allowed thresholds.

This means that the blocking probability for class $i$ calls offered to the access cell $z$ in the access cell $z$ ($B_{zz,i}^{UL}$) and the blocking probability for class $i$ calls offered to the access cell $z$ in the adjacent cell $h$ ($B_{zh,i}^{UL}$) depend on the traffic streams offered to the access cell $z$ and to the adjacent cells. Thus, we can express the above with the following functions:

$$B_{zz,i}^{UL} = f\left\{ \begin{array}{l} (a_{zz,1}, t_{zz,1}), \ldots, (a_{zz,M}, t_{zz,M}), \\ (a_{hz,1}, t_{hz,1}^{'}), \ldots, (a_{hz,M}, t_{hz,M}^{'}) \end{array} \right\}, \tag{15}$$

and

$$B_{zh,i}^{UL} = f\left\{ \begin{array}{l} (a_{hh,1}, t_{hh,1}), \ldots, (a_{hh,M}, t_{hh,M}), \\ (a_{zh,1}, t_{zh,1}^{'}), \ldots, (a_{zh,M}, t_{zh,M}^{'}) \end{array} \right\}. \tag{16}$$

The parameters $t_{hz,1}^{'}, \ldots, t_{hz,M}^{'}$ and $t_{zh,1}^{'}, \ldots, t_{zh,M}^{'}$ for class $i$ can be obtained as follows:

$$t_{zh,i}^{'} = L_{zh,i}^{'}/L_{hh,BBU} \quad \text{and} \quad t_{hz,i}^{'} = L_{hz,i}^{'}/L_{zz,BBU}, \tag{17}$$

where $L_{zz,BBU}$ and $L_{hh,BBU}$ can be determined on the basis of (7) for the cell $z$ and $h$:

$$L_{zz,BBU} = GCD(L_{zz,1}, \ldots, L_{zz,M}, L_{hz,1}^{'}, \ldots, L_{hz,M}^{'}), \tag{18}$$

$$L_{hh,BBU} = GCD(L_{hh,1}, \ldots, L_{hh,M}, L_{zh,1}^{'}, \ldots, L_{zh,M}^{'}). \tag{19}$$

The functions $B_{zz,i}^{UL}$ and $B_{zh,i}^{UL}$ can be determined on the basis of the FAG which serviced an infinite (Eq. (8) and (10)) and a finite (Eq. (13) and (14)) number of sources. A fixed-point methodology was used to determine traffic $a_{zh,i}^{UL}$ offered to the cell $h$ by a class $i$ call serviced in the $z$ cell. In keeping in this method, only such traffic which is blocked in the neighboring cells can be offered to a given cell. This phenomenon leads to a decrease in the traffic offered to a given cell and is called the thinning effect [11]. The class $i$ traffic stream, which is offered to the cell $h$ by a call occurring in the access cell $z$, decreased by the thinning effect, is called the effective traffic. Thus, parameters $a_{zh,i}^{UL}$ and $a_{zz,i}^{UL}$ can be obtained as follows:

$$a_{zh,i}^{UL} = a_{z,i}^{UL}\left(1 - B_{zz,i}^{UL}\right) \prod_{k=1, e_k \neq h}^{|\mathbb{S}_z|} \left(1 - B_{ze_k,i}^{UL}\right) \quad \text{and} \quad a_{zz,i}^{UL} = a_{z,i}^{UL} \prod_{k=1}^{|\mathbb{S}_z|} \left(1 - B_{ze_k,i}^{UL}\right), \tag{20}$$

where $a_{z,i}^{UL}$ is class $i$ traffic offered to the system by users in the $z$ cell in the uplink direction and $e_k$ is the element of the set of the neighboring cells for the cell $z$ ($\mathbb{S}_z$).
It should be noted that to determine the effective class $i$ traffic $a_{zh,i}^{UL}$, the information on the blocking probability $B_{zh,i}^{UL}$ of the traffic of this class in the neighboring cells is indispensable. Therefore, to determine the value $a_{zh,i}^{UL}$ the iterative method is used.

If we know the blocking probability $B_{zh,i}^{UL}$ of class $i$ call in the cell $h$ offered originally in the cell $z$, we can determine the blocking probability $B_{zz,i}^{UL}$ of class $i$ calls in the uplink direction of the cell $z$:

$$B_{zz,i}^{UL} = 1 - \prod_{k=1}^{|\mathbb{S}_z|} \left(1 - B_{ze_k,i}^{UL}\right) \quad \text{where} \quad e_k \in \mathbb{S}_z. \tag{21}$$

Consider now the system shown in Figure 1(b). The system services class $i$ calls that demand $t_i$ resources in each downlink direction of the cell to which they are offered (Eqs. (7 and (9)). The radio interface of each cell in the downlink direction can be described by the FAG. Thus, the blocking probability of class $i$ occurring in the downlink direction of the access cell ($z$) can be expressed by the function:

$$B_{zz,i}^{DL} = f\left\{ \ (a_{zz,1}, t_{zz,1}), \ldots, (a_{zz,M}, t_{zz,M}) \ \right\}. \tag{22}$$

The blocking probabilities $B_{zz,i}^{DL}$ can be determined on the basis of the FAG with an infinite (Eqs. (8) and (10)) and a finite (Eqs. (13) and (14)) number of sources. The service processes of calls of particular classes in the downlink direction are independent (Sect. 3.1.) and do not require taking into account dependencies between neighboring cells.

#### 3.3.1. Blocking probability in the group of cells

In our model we assume that, in the case of services demanding resources in both directions, the service processes in each direction are interdependent. We can take into account this dependency using the fixed-point methodology [11] which in Sect. 2. was used for determination of the value of blocking probabilities in the uplink direction. This methodology was used to determine traffic offered to the cell $z$ by a class $i$ call serviced in the $z$ cell in each of the directions. The class $i$ traffic stream, which is offered to the cell $z$ by a call occurring in the uplink and downlink directions of the cell $z$, can be obtained as follows:

$$a_{zz,i}^{UL} = a_{zz,i}(1 - B_{zz,i}^{DL}) \quad \text{and} \quad a_{zz,i}^{DL} = a_{zz,i}(1 - B_{zz,i}^{UL}), \tag{23}$$

where $a_{zz,i}$ is the total traffic offered by class $i$ calls to the cell $z$ and, the blocking probabilities $B_{zz,i}^{UL}$ and $B_{zz,i}^{UL}$ are calculated based on (21) and (22) respectively. Thus, the blocking probability of class $i$ call stream in the cell $z$ can be calculated using the iterative method, the same way as the blocking probabilities in the uplink direction, and can be determined by the following formula :

$$B_{zz,i} = 1 - (1 - B_{zz,i}^{DL})(1 - B_{zz,i}^{UL}). \tag{24}$$

Services which demand only one direction, are calculated by the appropriate method depending on the direction in which they are offered (Eqs. (21) and (22)).

## 4. Numerical results

In order to confirm the proposed method of blocking probability calculation in a cell with the WCDMA the radio interface and a finite and an infinite number of sources, the results of the analytical calculations were compared with the results of the simulation experiments. The study was carried out for users demanding a set of services (Table 1) and it was assumed that: a call of the particular services demanded $t_1 = 53$ (UL/DL), $t_2 = 257$ (UL/DL), $t_3 = 503/1118$ (UL/DL) and $t_4 = 1118/1118$ (UL/DL) BBUs, the holding time in the uplink

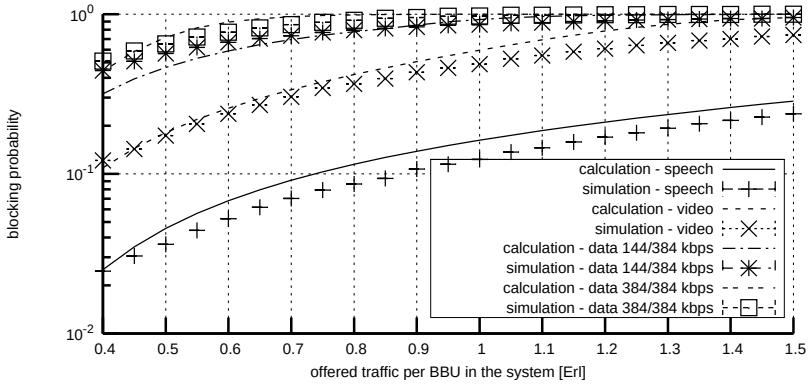



Fig. 3. Blocking probability in the group of cells for infinite number of sources, $d = 600$ meters.

and downlink direction for each service is the same (based on practical implementation of the UMTS system), the services were demanded in equal proportions (i.e. $a_1t_1 : a_2t_2 : a_3t_3 : a_4t_4 = 1 : 1 : 1 : 1$), the maximum theoretical uplink direction load capacity for each cell was equal to 10,000 BBUs, the $L_{BBU}$ in our model was equal to 0.0001 of the interface capacity and the maximum uplink and downlink directions load ($\eta_{UL/DL}$) were set to 50% of the theoretical capacity (i.e. $V = 5000$ BBUs for each cell in each direction)[12].

Figures 3 - 4 show the mean blocking probability of a new call as a function of the offered traffic for 4 traffic classes serviced by the system with an infinite and finite number of sources[1] The results presented in Figs. 3 and 4 confirm the known dependence according to which the increase in the number of the traffic sources is accompanied by the increase in the values of blocking probabilities, with other parameters of the system and the traffic offered being the same. The investigations were carried out for many numbers of traffic sources and geometrical size of the cells. Due to the limited length of the paper, we have selected exemplary results to present in the paper. All the presented results show the robustness of the proposed method of blocking probability calculation. In each case, regardless of the offered traffic load, the results are characterized by fair accuracy. The results of the simulations are shown in the charts (Figs. 3–4) in the form of marks with 95% confidence intervals that were calculated after the $t$-Student distribution. 95% confidence intervals of the simulation are almost included within the marks plotted in the figures.

## 5. Conclusions

The admission control in wireless networks with the WCDMA radio interface admits or blocks new calls depending on a current load situation both in the access cell and in

[1]The size cell influences the value of $L_i^{'}$ parameter and a longer distance between Node B's ($d$) effects in a higher load of the system and an increase in blocking probabilities. Due to limited length of the paper we present only the results of one exemplary plot (Fig. 4).

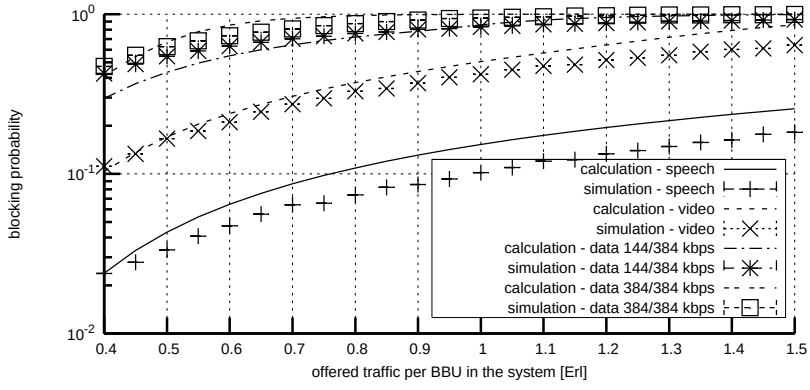



Fig. 4. Blocking probability in the group of cells for 800 sources $(200 : 200 : 200 : 200)$ and $d = 4800$ meters.

neighboring cells. In this article we present a blocking probability calculation method for cellular systems with the WCDMA radio interface for the uplink and the downlink direction. In our model we use load factor $L_i$ to estimate whether a new call of class $i$ service can be admitted or blocked. In the proposed calculation method the corresponding value of the load factor in the neighboring cell is based on the Okumura-Hata propagation model. The calculations are validated by a simulation. The proposed method can be easily applied to 3G network capacity calculations.

## References

[1] *Final Report*. European Communities, 1999.

[2] Saleh Faruque. *Cellular Mobile Systems Engineering*. Artech House, London, 1997.

[3] R. Fortet, C. Grandjean. Congestion in a loss system when some calls want several devices simultaneously. *Electrical Communication*, 39(4):513–526, 1964.

[4] M. Głąbowski, M. Stasiak. An approximate model of the full-availability group with multi-rate traffic and a finite source population, *Proc. of 3rd Polish-German Teletraffic Symposium*, pp. 195–204, Dresden, Germany, 2004. VDE Verlag.

[5] M. Głąbowski, M. Stasiak, A. Wiśniewski, P. Zwierzykowski. Uplink blocking probability calculation for cellular systems with wcdma radio interface, finite source population and differently loaded neighbouring cells. *Proc. of Asia-Pacific Conference on Comm.*, Australia, 2005.

[6] H. Holma, A. Toskala. *WCDMA for UMTS. Radio Access For Third Generation Mobile Communications*. John Wiley & Sons, Ltd., 2000.

[7] V.B. Iversen, editor. *Teletraffic Engineering Handbook*. ITU-D, Study Group 2, Question 16/2, Geneva, December 2003.

[8] V.B. Iversen, V. Benetis, H.N. Trung. Evaluation of multi-service CDMA networks with soft blocking. In *ITC 16th Specialist Seminar on Performance Evaluation of Mobile and Wireless Systems*, pp. 212–216, Antwerp, Belgium, 2004.

[9] V.B. Iversen, E.Epifania. Teletraffic engineering of multi-band W-CDMA systems. pp. 90–103, 2003.

[10] J.S. Kaufman. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 29(10):1474–1481, 1981.

[11] F.P. Kelly. Loss networks. *The Annals of Applied Probability*, 1(3):319–378, 1991.

[12] J. Laiho, A. Wacker, T. Novosad. *Radio Network Planning and Optimization for UMTS*. John Wiley & Sons, Ltd., 2006.

[13] Hata M. Empirical formula for propagation loss in land mobile radio services. *IEEE Trans. on Vehicular Technology*, 29:317–325, August 1980.

[14] Ohmori E. at al. Okumura Y. Field strength and its variability in vhf and uhf land-mobile radio service. *Rev. of the Electcrical Communication Laboratory*, 16(9):825–873, 1968.

[15] J.W. Roberts. A service system with heterogeneous user requirements — application to multi-service telecommunications systems, *Proc. of Performance of Data Communications Systems and their Applications*, pp. 423–431, 1981.

[16] J.W. Roberts, V. Mocci, I. Virtamo, editors. *Broadband Network Teletraffic, Final Report of Action COST 242*. Springer, Berlin, 1996.

[17] Lee J. S., Miller L. E. *CDMA Systems Engineering Handbook*. Artech House, 1998.

[18] D. Staehle, A. Mäder. An analytic approximation of the uplink capacity in a UMTS network with heterogeneous traffic. *Proc. 18th Int. Tele. Congress*, pp. 81–91, 2003.

[19] M. Stasiak, M. Głąbowski. A simple approximation of the link model with reservation by a one-dimensional Markov chain. *Performance Evaluation*, 41(2–3):195–208, 2000.

[20] M. Stasiak, A. Wiśniewski, P. Zwierzykowski. Blocking probability calculation in the uplink direction for cellular systems with WCDMA radio interface. *Proc. of 3rd Polish-German Teletraffic Symposium*, pp. 65–74, Germany, 2004.

[21] M. Stasiak, A. Wiśniewski, P. Zwierzykowski. Prawdopodbieństwo blokady dla łącza "w górę" w systemach komórkowych z interfejsem WCDMA. *Zeszyty Naukowe Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej*, 1(1), 2007.

[22] A.M. Viterbi, A.J. Viterbi. Erlang capacity of a power controlled CDMA system. *IEEE Journal on Selected Areas in Communications*, 11(6):892–900, 1993.

[23] H. Holma, A. Toskala. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. John Wiley & Sons, 2006.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 111–120

# An Algorithm for Computing the Blocking Probabilities in Cellular Mobile Communication Networks

JERZY MARTYNA[a]

[a]Institute of Computer Science Jagiellonian University
ul. Nawojki 11, 30-072 Cracow, Poland
martyna@softlab.ii.uj.edu.pl

**Abstract:** In this paper we introduced an algorithm for calculating the steady-state blocking probabilities in cellular mobile communication networks. In these networks, the blocking probabilities have simple expressions in terms of normalization constants. Assuming that a model of a cellular mobile communication networks contains their product-form solution, we formulated the steady-state blocking probability of each traffic class in terms of normalization constants. To calculate the blocking probabilities, we computed the normalization constant. We also gave some numerical examples which illustrate our method.

**Keywords:** Blocking probabilities, cellular mobile communication networks, convolution algorithm

## 1. Introduction

Cellular mobile communication networks require investing effort with both improving QoS and ensuring their reliability. However, it is impossible to obtain standard measures and tools to evaluate the performance of a system. The Erlang model and several formulas are often used to design circuit-switched networks, but such a model must be intractable for mobile communication networks.

One of the first research studies of mobile communication networks was presented be Everit and Manfield [1]. However, their model was not treated as a queueing network model. A paper by Boucherie and Mandjes [2] specified the equilibrium distribution for product form cellular mobile communication networks involving a single call class. As representative research, Yoneyama et al. [3] presented a queueing network model with multiple call classes and state-dependent transition rates with derived product form equilibrium distribution. However, in both papers no channel was available if the call was lost. This is referred to as so-called "handoff" blocking.

Algorithms for computing blocking probabilities in terms of normalization constants were only developed for special cases. Kaufman [4] and Roberts [5] used recursion for loss

networks. Pinsky and Conway [6] used this method in computation algorithms for blocking probabilities in Circuit-Switched Networks. Mitra [7] adopted this method for a class of two-hop tree networks. Kogan [8] developed an algorithm for this model by relating it to closed networks.

In this paper, we introduce a new method for numerically computating blocking probabilities in cellular mobile communication networks. We formulate a model of these networks as a queueing network with multiple call classes and state-dependent transition rates. To measure performance we suggest a handoff blocking probability based on a product form equilibrium distribution. We propose an algorithm for computing these blocking probabilities. As numerical example, we investigated the blocking probabilities between neighbouring cells in a nine-cell model.

## 2. A Model of a Cellular Mobile Communications Network and its Product-Form Expressions

Consider a cellular mobile communications network consisting of $N$ cells and $I$ classes of calls. The number of class $u$ calls in cell $i$ is denoted by $n_{ii(u)}$ and the number of class $u$ calls in the handover area between cells $i$ and $j$ is denoted by $n_{ij(u)}$. Thus, the number of class $u$ calls carried by transceiver $i$ is given by

$$m_{i(u)} = n_{ii(u)} + \sum_{j \in H_i} n_{ij(u)}, \quad u = 1, 2, \ldots, I \tag{1}$$

where $H$ is a set of neighbours of cell $i$. In general

$$m_i = n_{ii} + \sum_{j \in H_i}, \quad i = 1, 2, \ldots, N \tag{2}$$

where $m_i = \sum_{u=1}^{I} m_{i(u)}$, $n_{ii} = \sum_{u=1}^{I} n_{ii(u)}$, $n_{ij} = \sum_{u=1}^{I} n_{ij(u)}$. We assume that a new call class $u$ is generated by a Poisson process in cell $i$ with the rate $\lambda_{ii(u)}$, $i = 1, 2, \ldots, N$, $u = 1, 2, \ldots, I$, and in the handoff area between cells $i$ and $j$ with $\lambda_{ij(u)}$, $i, j = 1, 2, \ldots, N$, $u = 1, 2, \ldots, I$. We suppose that the mobile class $u$ call remains in cell $i$ for a holding time exponentially distributed with the mean $\mu^{\star}_{ii(u)}$. The class $u$ call can enter the handoff area with cell $j$ as a class $v$ call (to the handoff area with cell $i$) with probability $p_{ii(u),ij(v)}$. The class $u$ call can end in cell $i$ or in the handoff area $ij$ with the rate $\mu'_{ii(u)}$ or $\mu'_{ij(u)}$, respectively.

The holding time of a class $u$ call in the cell $i$ is exponentially distributed with the mean $1/(\mu'_{ii(u)} + \mu^{*}_{ii(u)})$. The probability that class $u$ call ends in cell $i$ is equal to $p_{ii(u),0} = \mu'_{ii(u)}/\mu_{ii(u)}$. Analogously, the probability that a class $u$ call will enter the handoff area with cell $j$ as a class $v$ call is equal to $p_{ii(u),ii(v)} = \frac{\mu'_{ii(u)}}{\mu_{ii(u)}} p_{ii(u),ij(v)}$. Two conditions are satisfied, namely

$$\sum_{j \in H_i} p^*_{ii(u),ij(v)} = 1 \tag{3}$$

$$p_{ii(u),0} + \sum_{j \in H_i} p_{ii(u),ij(v)} = 1 \tag{4}$$

A class $u$ call remains in the handoff area between cell $i$ and $j$ for $\exp(\mu_{ij(u)})$ exponentially distributed with the mean $\mu_{ij(u)}$.

A class $u$ call moves or returns to the interior of cell $i$ as a class $v$ call with probability $p_{ij(u),ij(v)}$. The handoff of a class $u$ as a class $v$ call is attempted with probability $p_{ij(u),ji(v)}$.

A cellular mobile communications network can be modeled as continuous time Markov chain **X** = $(X(t), t \geq 0)$, which contains the number of calls in the areas of all cells. The state of this Markov chain is given by vector **n** = ($\mathbf{n}_{ii}$, $\mathbf{n}_{ij}$, $j \in H_i$, $i = 1, 2, \ldots, N$), where $\mathbf{n}_{ii}$ = $(n_{ii(1)}, \ldots, n_{ii(I)})$, $\mathbf{n}_{ij}$ = $(n_{ij(1)}, \ldots, n_{ij(I)})$. We assume that the total number of calls in the cells is given by $m = (m_1, \ldots, m_N)$, where $m_i = \sum_{u=1}^{I} m_{i(u)}$. All restrictions are inputed on vector $m$. Nevertheless, our model of cellular mobile communications networks includes both fixed and dynamic channel allocation schemes.

We assume that the Markov chain is irreducible and its state space is given by **S** = {**n**: **Am** $\leq$ **C**, $m_i = n_{ii} + \sum_{j \in H_i} n_{ij}, i = 1, 2, \ldots, N\}$, where **A** is a matrix with nonnegative entries, **C** is a vector of constraints.

A product form expression for the cellular mobile communications network with infinite capacity is obtained by assuming the partial balance of the network.

Thus, we can determine the traffic equation for all class $u$ calls in the cell $i$ as follows

$$c_{ii(u)} = \lambda_{ii(u)} + \sum_{j \in H_i} \sum_{v=1}^{I} p_{ij(v),ii(u)} \tag{5}$$

$$c_{ij(u)} = \lambda_{ij(u)} + \sum_{v=1}^{I} c_{ii(v)} \cdot p_{ii(v),ij(u)} + \sum_{v=1}^{I} c_{ji(v)} \cdot p_{ji(v),ij(u)} \tag{6}$$

where $c_{ii(u)}$ is the arrival rate of class $u$ calls in cell $i$ and $c_{ij}$ is the arrival rate of class $u$ calls in the handoff area $ij$.

The rate $\mu_{ii(u)}$, $i = 1, 2, \ldots, N$, $u = 1, 2, \ldots, I$ for the state of network **n** is given by

$$\mu_{ii(u)}(\mathbf{n}) = \mu_{ii(u)} \frac{\Psi(\mathbf{n} - e_{ii(u)})}{\Phi(\mathbf{n})} \tag{7}$$

where $\Psi$ and $\Phi$ are arbitrary nonnegative and positive functions, $e_{ii(u)}$ is a row vector with 1 in $ii$ place and 0's elsewhere.

The rate $\lambda_{ii(u)}$ for $i = 1, 2, \ldots, N$, $u = 1, 2, \ldots, I$ for the state of network **n** is as follows

$$\lambda_{ii(u)}(\mathbf{n}) = \lambda_{ii(u)} \frac{\Psi(\mathbf{n})}{\Phi(\mathbf{n})} \tag{8}$$

For $j \in H,\ i = 1, 2, \ldots, N,\ u = 1, 2, \ldots, I$ the rate $\mu_{ij(u)}$ when the state of the network $\mathbf{n}$ is described by

$$\mu_{ij(u)} = \lambda_{ii(u)} \frac{\Psi(\mathbf{n} - e_{ij(u)})}{\Phi(\mathbf{n})} \tag{9}$$

where $e_{ij(u)}$ is a row vector with 1 in the $ij$ place and 0's elsewhere. Analogously, when the state of the network is given by $\mathbf{n}$, the rate $\lambda_{ij(u)}$ can be formulated as follows

$$\lambda_{ij(u)}(\mathbf{n}) = \lambda_{ij(u)} \frac{\Psi(\mathbf{n})}{\Phi(\mathbf{n})} \tag{10}$$

The product form expressions for a cellular mobile communications network with multiple call classes and state-dependent transition is given by

$$\begin{aligned} \pi(\mathbf{n}) \quad = \quad & \frac{1}{G(C)} \Phi(\mathbf{n}) \prod_{i=1}^{N} \prod_{j \in H_i} \frac{1}{n_{ii(1)}!, \ldots, n_{ii(I)}!} \cdot \frac{1}{n_{ij(1)}!, \ldots, n_{ij(I)}!} \\ & \cdot \prod_{v=1}^{I} \left(\frac{c_{ii(v)}}{\mu_{ii(v)}}\right)^{n_{ii(v)}} \left(\frac{c_{ij(v)}}{\mu_{ij(v)}}\right)^{n_{ij(v)}} \end{aligned} \tag{11}$$

where $G(C)$ is the normalizing constant.

The basic formula for the blocking probability for class $u$ calls at handoff area $ij$ is given by

$$B_{ij(u)} = 1 - G(C - Ae'_{ij(u)})/G(C) \tag{12}$$

where $e'_{ij(u)}$ is the transpose of a matrix with a 1 in the $ij$-th place and 0's elsewhere.

Now, we assume that the upper limit $L_{ij(u)}$ is imposed on class $u$ calls at handoff area $ij$. Thus, the condition $\lfloor L_{ij(u)}/a_{ij(u)} \rfloor \geq \lfloor L'_{ij(u)}/a'_{ij(u)} \rfloor$ for all $i' = 1, \ldots, N'$, where $N' \geq N$. We note that $Am' \geq C$, $m' \geq D$, where $D$ is the set of call classes which are carried by transceiver $i$. Let $G(C, D)$ be the normalization constant as a function of the pair $(C, D)$. Thus, the blocking probability for class $u$ calls at handoff area $ij$ is given by

$$B_{ij(u)} = 1 - \frac{G(C - Ae'_{ij(u)}, D - Ae'_{ij(u)})}{G(C, D)} \tag{13}$$

If $D$ represents the final class $u$ calls in the ordering this becomes simply

$$B_{ij(D)} = 1 - \frac{G(C - A \cdot e'_{ij(D)}, D - 1)}{G(C, D)} \tag{14}$$

where $G(C, D)$ is the normalization constant.

## 3. An Algorithm for Computing the Blocking Probabilities in Cellular Mobile Communications Networks

The computational algorithms which are available deal with closed product form networks. The algorithms generally calulate blocking probabilities for networks with $N$ customers based on the previous results for $N-1$ customers. One approach, due independently to Buzen [9] and Reiser [10, 11], is called the *convolution algorithm*. It is basically a recursion for the normalization constants for increasing customer populations.

Direct computation of a normalizing constant requires summing over the entire state space. The number of states is the same as the number of ways that $D_u$ classes can be arranged over $C$ constraints (for each class $u$ call). Thus, direct computation requires

$$\prod_{u=1}^{U} \begin{pmatrix} D_u + C - 1 \\ C - 1 \end{pmatrix} \text{ steps}$$

where $U$ is the total number of calls.

The basic case and recurrence for computing the blocking probabilities are given by

$$B_{ij(u)} = 1 - \frac{G(C - A \cdot e'_{ij(u)}, D - A \cdot e'_{ij(u)})}{G(C, D)} \tag{15}$$

The numerator can be computed using the previous recursive expression for the auxiliary function. If $D$ is the last class $u$ in the ordering this becomes simply:

$$B_{ij(D)} = 1 - \frac{G(C - A \cdot e'_{ij(D)}, D - 1)}{G(C, D)} \tag{16}$$

If term $G(C - A \cdot e'_{ij(D)}, D - 1)/G(C, D)$ in Eq. (16) enables the use of a recursion to compute the blocking probabilities. The blocking probabilities in a cellular mobile communications system are computed by the *blocking_prob._computation* procedure. The convolution algorithm used in this procedure is given as follow:

**Algorithm 1**

**Step 1:** Define a matrix $G(C - Ae'_{ij(u)}, D - Ae'_{ij(u)})$ of size $(u+1)U$, where $U$ is the number of class calls.

**Step 2:** All the elements in the first row of the $G$-matrix are 1, since $G_u(0) = 1$, for $u = 1, 2, \ldots, U$.

**Step 3:** All the elements in the first column of the $G$-matrix are calculated by the formula

$$G_1(j) = x_1 G_1(j-1) = x_1^j$$

**Step 4:** All the elements from row 2 to $U$ and columns 2 to $u+1$ are determined by the calculation

$$g(k,l) = g(k,l-1) + g(k-1,l)x_l, \;\; k = 1,2,\ldots,U, \;\; l = 1,2,\ldots,U$$

**Step 5:** When the calculation process is complete, the $G$ coefficients are the elements of last column.

**Step 6:** Computation of the handoff blocking probability

$$B_u(k) = 1 - G_u(k), k = 1,\ldots,U$$

■

## 4. Numerical Examples

Based on the algorithm for computing handoff blocking probabilities, we studied its usefulness as a tool for analysing cellular mobile communication networks. We considered a cellular network set consisting of nine cells (see Fig. 1). In our model we investigated the two-way connection of its immediate neighbours. We treat the handoff blocking probability as a function of a cell, say $L_i$, when the connection ratio $A_{ij}$ is given. The basic parameters of constraint matrix are shown in Table 1.

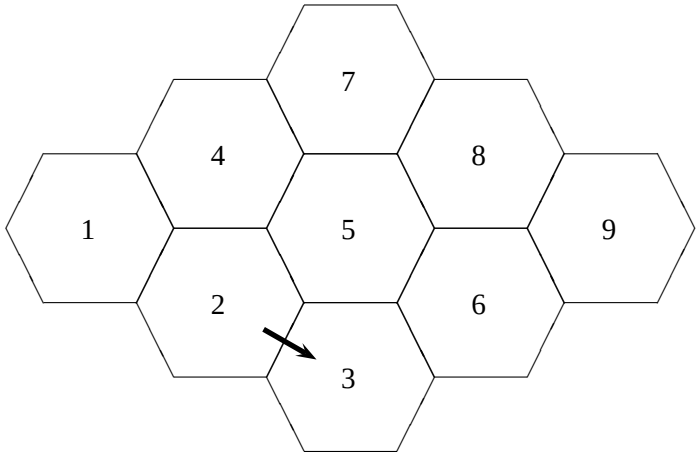



**Fig. 1.** Nine-cells layout.

In our study we investigated handoff blocking probabilities as a function of the load of cell 3 for three several connection ratios, namely $A_{23}$ = 0.35, 0.25, 0.15. We assumed that

the other connection ratios, $A_{ij}$, have the same value. This means that the ratio of a two-way connection to calls is equally distributed in all cells. The computed handoff blocking probability between cell 2 and 3 for the load in cell 3 is given in Fig. 2. It can be seen that the handoff blocking probability between cell 2 and 3 increases when load $L_3$ is increased. This is due to the load which is increased in cell 3.

**Table 1.** Basic constraint matrix.

| Cells | Cells | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 1.0 | 0.2 | 0.0 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.2 | 1.0 | 0.2 | 0.2 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.2 | 1.0 | 0.0 | 0.2 | 0.2 | 0.0 | 0.0 | 0.0 |
| 4 | 0.2 | 0.2 | 0.0 | 1.0 | 0.2 | 0.0 | 0.2 | 0.0 | 0.0 |
| 5 | 0.0 | 0.2 | 0.2 | 0.2 | 1.0 | 0.2 | 0.2 | 0.2 | 0.0 |
| 6 | 0.0 | 0.0 | 0.2 | 0.0 | 0.2 | 1.0 | 0.0 | 0.2 | 0.2 |
| 7 | 0.0 | 0.0 | 0.0 | 0.2 | 0.2 | 0.0 | 1.0 | 0.2 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 1.0 | 0.2 |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.0 | 0.2 | 1.0 |

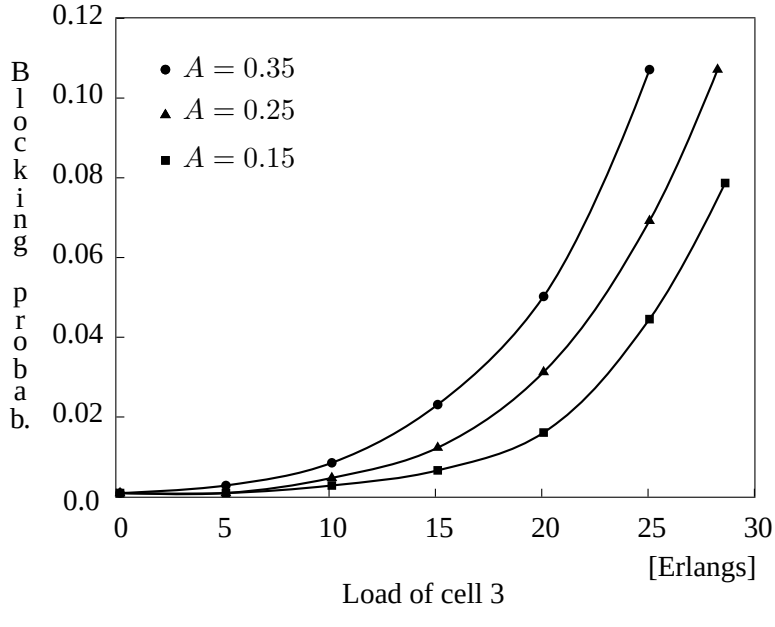



**Fig. 2.** The handoff blocking probability as a function of load $L_3$.

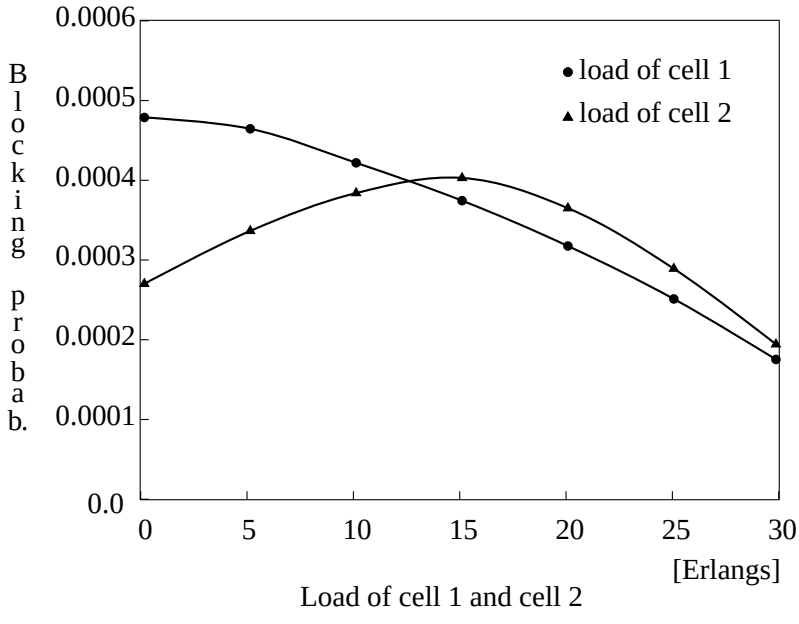



**Fig. 3.** The handoff blocking probability between cell 2 and 3 as a function of load of cell 1 and cell 2.

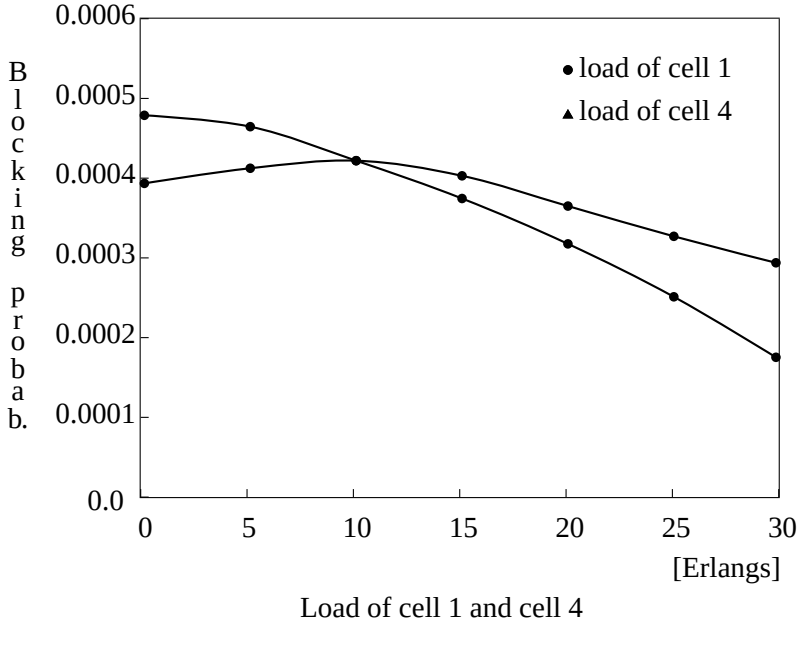



**Fig. 4.** The handoff blocking probability between cell 2 and 3 as a function of load of cell 1 and cell 4.

The handoff blocking probabilities between cells 2 and 3 as a function of load in cell 1 and 2 are given in Fig. 3. Because the cell 1 is not the immediate neighbour of cell 3, we argue that the load of cell 1 has no influence on handoff calls from cell 2 and 3. Figure 4 shows the handoff blocking probabilities between cell 2 and 3 as a function of load in cell 1 and cell 4. It can be seen that the handoff blocking probability decreases as $L_{14}$ increases. In other words the loads of cells 1 and 4 have no influence on blocking probability in cell 3.

## 5. Conclusions

In this paper, we proposed an algorithm for computing the blocking probabilities in a cellular mobile communications system. This algorithm is based on the product form queueing network model and uses a convolution technique. In our model we can give spatial information about the load of a cell in a cellular system.
We can make our algorithm useful for planning control in cellular mobile communications networks as well as for quantitative analysis by investigating the behavior of the system.

## References

[1] D.E. Everitt, D. Manfield, "Performance Analysis of Cellular Mobile Communication Systems with Dynamic Channel Assignment", IEEE Journal on Selected Areas in Communications, **8**, 1989, pp. 1172-1180.

[2] R.J. Boucherie, M. Mandjes, "Estimation of Performance Measures for Product Form Cellular Mobile Communicationd Networks", Telecommunication Systems, **10**, 1998, pp. 321-354.

[3] K. Yoneyama, H. Kawane, H. Ishii, "Product Form Equilibrium Distribution for Cellular Mobile Communications Network with Multiple Call Classes and State-Dependent Transitions", Technical Report of IEIC, IN2001, 2001, pp. 41-47.

[4] J.S. Kaufman, "Blocking in a Shared Resource Environment", IEEE Transactions on Communications, **COM-29**, 1981, pp. 1474-1481.

[5] J.W. Roberts, "Service System with Heterogeneous User Requirements", Performance of Data Communications Systems and Their Applications, G. Pujolle (Ed.), North-Holland Publishing, 1981, pp. 423-431.

[6] E. Pinsky, A.E. Conway, "Computational Algorithms for Blocking Probabilities in Circuit-Switched Networks", Annals of Operations Research, 1992, pp. 31-41.

[7] D. Mitra, "Asymptotic Analysis and Computational Methods for a Class of Simple, Cicuit-Switched Networks with Blocking", Advances in Applied Probability, **19**, 1987, pp. 291-239.

[8] Y. Kogan, "Exact Analysis for a Class of Simple, Circuit-Switched Networks with Blocking", Advances in Applied Probability, **21**, 1989, pp. 952-955.

[9] J.P. Buzen, "Computational Algorithms for Closed Queueing Networks with Exponential Servers", Communications of the ACM, **16**, 9, 1973, pp. 527-531.

[10] M. Reiser, H. Kobayashi, "Recursive Algorithms for General Queueing Networks with Exponential Servers", IBM Research Report, RC 4254, Yorktown Heights, N.Y., 1973.

[11] M. Reiser, "Mean-Value Analysis and Convolution Method for Queue-Dependent Servers in Closed Queueing Networks", Performance Evaluation, **1**, 1, 1981, pp. 7-18.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 121–130

# Modeling the Lifetime of Hierarchical Wireless Ad Hoc and Sensor Networks

JERZY MARTYNA[a]

[a]Institute of Computer Science Jagiellonian University
ul. Nawojki 11, 30-072 Cracow, Poland
martyna@softlab.ii.uj.edu.pl

**Abstract:** The lifetime of a wireless ad hoc and sensor network is defined as the time after which a certain fraction of nodes exhaust their batteries power. We propose a new mathematical model for determining the lifetime of these networks. In our approach we have taken into consideration data loss within the network as well as the damage to a node or the lack of energy for data transmission. so as to achieve that a maximum lifetime. The effectiveness of this lifetime model is outlined by the simulation experiments.

**Keywords:** wireless ad hoc networks, wireless sensor networks, model of lifetime

## 1. Introduction

*Ad hoc networks* [1, 2] consist of nodes that may be mobile and have wireless communication capability without the benefit of mediating infrastructure. Every node can become aware of the presence of others within its radio range. Such nodes communicate directly by their wireless communication links. Neighbours can communicate directly with one other. And every node also plays an active role in forwarding data units to other nodes.

*Wireless sensor networks* (WSNs) [3] are a particular type of ad hoc network, in which the nodes are small devices equipped with advanced sensing functionalities (pressure, acoustic, etc.), a small processor and a short-range wireless transceiver. In this type of network, the sensors exchange information on the environment in order to build a global view of the monitored region.

The *lifetime* of a wireless sensor network as well as an ad hoc network is defined as the after which the first node (link) disconnects [4, 5]. This definition is especially useful for determining the lifetime of these networks functioning in real-time applications. It is also referred to as the *worst-case lifetime* model.

The lifetime model of sensor networks and ad hoc networks was developed in study by Bhardwaj [6], and in a number of papers [4, 7, 8, 9, 10]. The upper limit of network lifetime when the data source and data rate are known was derived by Bhardwaj [6]. A distributed

procedure to find such capacity achieving routes was proposed in [4]. In other papers [7, 8] energy aware routing schemes were considered. Recently, minimizing the transmission cost [9] and the placement of nodes for energy efficiency [10] have been studied. However, none of the above given models provide an analysis of lifetime in hierarchical sensor networks achieved by clustering.

The main goal of this paper is to introduce a new framework for modeling the lifetime of WSNs and ad hoc networks. The first goal of this paper is to build a lifetime model of a single sensor node (SN), a single clusterhead, a cluster, and a whole WSN. The second goal of this paper is to answer the following question: is the lifetime of a WSN better for a flat WSN or two-tier hierarchy in WSN? And also, how are the lifetimes of a sensor node and a cluster in a WSN as a function of connectivity?

In the next section we discuss the lifetime of an SN. In Section 3, we introduce a framework for achieving a lifetime of sensors clusters, clusterhead sensors and a whole WSN. We give some illustrations of our methodology in Section 4. We conclude in Section 5.

## 2. The Lifetime of a Sensor Node

In this section, we provide a definition of the lifetime of a sensor node (SN) in WSNs. Since sensor networks are a subclass of wireless ad hoc networks, this lifetime model applies in this type of networks also.

We have made some assumptions in our framework:

1) All the technical parameters (such as initial energy power stored in the battery, the radio transmitting range, RAM memory, etc.) are the same for all nodes in WSN,
2) The radio transmitting range of SNs is limited to their neighbors,
3) All SNs have a fixed location in the sensor field. The sink is immovable,
4) SNs can be located by physical devices or topology discovery algorithms,
5) Communications between sensors run in a two-way direction within a communication range due to limited energy power.

We assume that each node in the WSN has finite energy $E_{bat}^{init}$. The battery energy is consumed in receiving and sending messages. When the battery energy is less or equal to $E_{bat}^{*}$, the node can be considered as "dead". Thus, based on the above given assumptions, it is possible to determine the lifetime of a single node, namely.

**Definition 1**

The lifetime of a continuously active SN is equal to

$$\tau_{life}^{sensor} = \frac{E_{bat}}{P_1} \tag{1}$$

where $P_1$ is the transmit power, $E_{bat}$ is the current energy power stored in the battery.

For a given spatial energy density in the network $\rho_S$, and $P_1$ we can give the maximum acceptable data-rate, $R_b^{max}$, namely

$$R_b^{max} = \frac{P_1 \cdot \rho_S}{\rho_{energy}^{min}} \tag{2}$$

where $\rho_{energy}^{min}$ is the minimum spatial energy density required for full connectivity. Thus, the minimum required transmit power for full connectivity at a given data-rate $R_b$ can be written as

$$P_t^{min} = \frac{\rho_{energy}^{min} \cdot R_b}{\rho_S} \tag{3}$$

The above relationship indicates that for a fixed number of nodes, if the nodes are damaged during the elapsed time, the minimum transmitting power must increase proportionally to preserve full connectivity in the network. This relationship also implies that when the node spatial density decreases and the minimum required transmit power $P_t$ is constant, then the data-rate $R_b$ must to be reduced.

Additionally, if $R_b$ and $P_1$ are fixed, the critical minimum node spatial density for full connectivity in WSN can be written as

$$\rho_S^{min} = \frac{\rho_{energy}^{min} \cdot R_b}{P_1} \tag{4}$$

Now, we can provide a better definition of the lifetime of a SN in a WSN.

**Definition 2**

When the data-rate $R_b$ and spatial energy density $\rho_S$ in the WSN are fixed, the lifetime of a node in the WSN can be formulated as

$$\tau_{life}^{sensor} = \frac{E_{bat} \cdot \rho_S}{R_b \cdot \rho_{energy}^{min}} \tag{5}$$

or

$$\tau_{life}^{sensor} = \frac{(E_{bat}^{init} - t \cdot e^{cons})\rho_S}{R_b \cdot \rho_{energy}^{min}} \tag{6}$$

where $E_{bat}^{init}$ is the initial energy power stored in the battery, $e^{cons}$ is the rate of energy consumed by a sensor, and $t$ is the elapsed time.

Note that a reduction in the data-rate will increase the lifetime of a continuously active sensor and thus the lifetime of a whole WSN.

After transforming the last equation, we can obtain the relationship for the *normalized lifetime* of SN, NLS. The normalized lifetime of an SN is defined as the normalized remaining energy of the sensor at the moment $t$, namely

$$NLS = \frac{\tau_{life}^{sensor}}{E_{bat}^{init}} = \frac{(1 - t \cdot e_{rel}^{cons})\rho_S}{R_b \cdot \rho_{energy}^{min}} \tag{7}$$

where $e_{rel}^{cons}$ is the relative energy consumption of a sensor node and is equal to the ratio $\frac{e^{cons}}{E_{bat}^{init}}$. Thus shows the relative speed of the energy consumed by a SN.

To calculate the energy consumption of each message transmission, we borrow the energy model used in [11]. The energy consumed when the sensor receives a message of size $k$ is given by

$$\epsilon_{rec} = \epsilon_{elec} \cdot k \tag{8}$$

where $\epsilon_{elec}$ is the energy consumed per one received bit. The energy consumed on sending a message of size $k$ is given by

$$\epsilon_{send} = \epsilon_{elec} \cdot k + \epsilon_{amp} \cdot r^2 \cdot k \tag{9}$$

where $\epsilon_{elec}$ is the consumed energy per one sent bit, $\epsilon_{amp}$ is the consumed energy by an amplifier, $r$ is the radio transmitting range of a single SN.

In this way, we obtain two amounts of energy consumed in one query message at the moment $t$, namely

$$\epsilon_q^{(t)} = \epsilon_{q,rec}^{(t)} + \epsilon_{q,send}^{(t)} \tag{10}$$

$$\epsilon_r^{(t)} = \epsilon_{r,rec}^{(t)} + \epsilon_{r,send}^{(t)} \tag{11}$$

where $\epsilon_r^{(t)}$ is the energy consumed in receiving, and $\epsilon_q^{(t)}$ is the energy consumed in sending. When we assume that all query messages are the same, $\epsilon_q^{(t)}$ and $\epsilon_r^{(t)}$ can be reduced to $\epsilon_q$ ($\epsilon_r$).

According to the results given in a paper by Chen [12], the energy consumption ratio when a SN is in idle mode, receiving mode and sending mode is: $1 : 1.2 : 1 : 1.68$. Therefore, the sending mode is the most exhaustive in terms of power consumption.

We can also determine the lifetime of a continuously active SN in more simpler form. We assume that for a given data-rate $R_b$ the time taken to transmit one packet (message) is $L/R_b$, where $L$ is the mean packet length. The total amount of energy consumed per transmitted packet (message) can be given by

$$E_{packet} = P_1 \cdot \frac{L}{R_b} (\text{dimension [J]}) \tag{12}$$

Let the transmission of a packet with an average rate be $\lambda_p$ and the average energy consumed per second $\lambda_p \cdot E_{packet}$. Thus, the total time required to completely exhaust the initial battery energy and thereby the lifetime of a SN is given by

$$\tau_{life}^{sensor} = \frac{E_{bat}^{init}}{\lambda E_{packet}} = \frac{E_{bat}^{init} R_b}{\lambda \cdot P_1 \cdot L} \text{ (dimension [s])} \tag{13}$$

We now point out that the above results are obtained by using a uniform traffic assumption (i.e. all nodes generate the same amount of traffic load) and all nodes consume the same energy per packet whether an SN is receiving or sending packets.

## 3. Determining the lifetime of clusterhead, cluster and fully hierarchically organized WSN and ad hoc networks

In this section, we will determine the lifetime of hierarchical organized WSNs and ad hoc networks.

WSN and ad hoc networks are often divided into groups defined as clusters. This concept was introduced by Baker et al. [13]. Such an approach allows us to manage a cluster and relay the collected data. In a paper by Ramamoorthy et al. [14] an expanding ring approach was used in which the cluster depth is progressively relaxed until the desired cluster size is exceeded. A hierarchy of networks is achieved by clustering technique. The first level of this hierarchy consists of SNs which "control" other SNs. They are often referred to as *clusterheads*. These nodes are natural places to aggregate and compress traffic data from many sensors. The communication between the clusterheads is dominating and is treated as *backbone* transmission. This communication is suitable for data transmission to the sink of a WSN.

In hierarchical WSNa and ad hoc networks all SNs are assigned to one of the clusters. There can be two ways of building clusters in a WSN. In the first option each cluster must be at most two hops away from any other node. In the second, one-hop clusters are considered. Since the battery of clusterheads will tend to be exhausted more quickly, it is desirable that all SNs in a cluster have equal battery capacities at any point. Therefore, the clustering algorithm should be able to rotate the clusterheads.

If there is a lack of rotating clusterheads, we propose here the following definition.

**Definition 3**

The lifetime of a clusterhead in the hierarchical ad hoc/WSN in case of lack of rotating clusterheads is given by

$$\tau_{life}^{clshead} = \frac{(E_{bat}^{init} - t \cdot e_{rel}^{cons})\rho_S}{R_b \cdot \rho_{energy}^{min} \cdot (c_1 \cdot r^{\alpha} + c_2 \cdot n_s)} \tag{14}$$

where $r$ is the coverage radius of a clusterhead, $n_s$ is the number of cluster members, $\alpha$ is the path-loss coefficient, and $c_1$ and $c_2$ are constants.

It can be seen that the maximization the hierarchical network lifetime involves maximizing the lifetime of all clusterheads or maximizing the minimum lifetime over all clusterheads.

Now we can formulate determine the lifetime of a cluster in hierachical WSN/ad hoc networks, namely

**Definition 4**

The lifetime of a cluster in an hierarchical ad hoc network/WSN can be written as

$$\tau_{life}^{cluster} = \frac{1}{N_c} \sum_{i=1}^{N_c} w_i^{(1)} \cdot \tau_{life,i}^{sens} \tag{15}$$

where $N_c$ is the total number of SNs at the moment when the network starts and $w_i^{(1)}$ is the weight of each node stated as follows

$$w_i^{(1)} = \frac{c_3}{d_i^2} \tag{16}$$

where $d_i$ is the distance from the $i$-th node to the clusterhead of a cluster.

Naturally, for both of the above given lifetimes we can form the normalized lifetimes of a clusterhead and a cluster.

Before we formulate the lifetime definition an entire hierarchical ad hoc and WSN network, we recall that there is is some difference between both of them. An ad hoc network is built to transport a data. On the other hand, a WSN is not designed to transport data, but rather to observe a region. Therefore, the WSN should be able to fulfill its duty for as long as possible.

There are a number of possible various definitions of an WSN lifetime.

1) The time until the first node fails, which is used in Real-Time WSNs,
2) The time until 50% of the nodes run out of energy and become inactive. Any other fixed percentile is applicable as well,
3) The time until there is a spot in an WSN that is not covered by the network,
4) The time until a network position (when there are two nodes in that a WSN can no longer communicate with each other) [15].

The need to maximize the lifetime of a WSN improves network performance. It means, that each of the given lifetime definitions requires different solutions. Nevertheless, we propose here two suitable definitions of flat and hierachical WSN and ad hoc network lifetimes.

For a flat WSN and ad hoc network (i.e. networks without a two- or multi-level hierarchy), we can give the following definition.

**Definition 5**

The lifetime of a flat ad hoc/WSN network is given by

$$\tau_{life}^{network} = \frac{1}{N} \sum_{i=1}^{N} w_i^{(2)} \cdot \tau_{life,i}^{sensor} \tag{17}$$

where $N$ is the total number of SNs in a WSN at the moment $t = 0$ and $w_i^{(2)}$ is the weight of each node, namely

$$w_i^{(2)} = \frac{c_4}{d_i^2} \tag{18}$$

where $d_i$ is the distance from the $i$-th node to the sink of the network.

The above definitions not consider a start up of a spot of nodes without transmitting range, as well as a possible a lack between two neighboring nodes.

For the two- (or multi) level network hierarchy, we can formulate the following definition.

**Definition 6**

The lifetime of a two-layer hierarchical WSN/ad hoc network is given

$$\tau_{life}^{network} = \frac{1}{K} \sum_{k=1}^{K} w_k^{(3)} \cdot \tau_{life,k}^{clusterhead} \tag{19}$$

where $K$ is the total number of clusterhead nodes, $w_k^{(3)}$ is the weight of each clusterhead node given as as follows

$$w_k^{(3)} = \frac{c_5}{d_k^2} \tag{20}$$

where $d_k$ is the distance from clusterhead $k$ to the sink of a WSN.

## 4. Numerical Experiments

In this section, we give some numerical results of lifetime modelling in hierarchical ad hoc and sensor networks.

We assumed that in our model of ad hoc and WSN the node spatial density of a network can change over time. Among other things, SNs are lost, when the batteries are exhausted. A WSN changes its node spatial density. We define the initial density of SNs when the network starts as $N_0$. Keeping the sensor field constant, the node spatial density is decreased if the number of SNs becomes lower than $N_0$. The change in the number of SNs is defined as $\Delta N = N_0 - N_f$, where $N_f$ is the the final number of SNs.

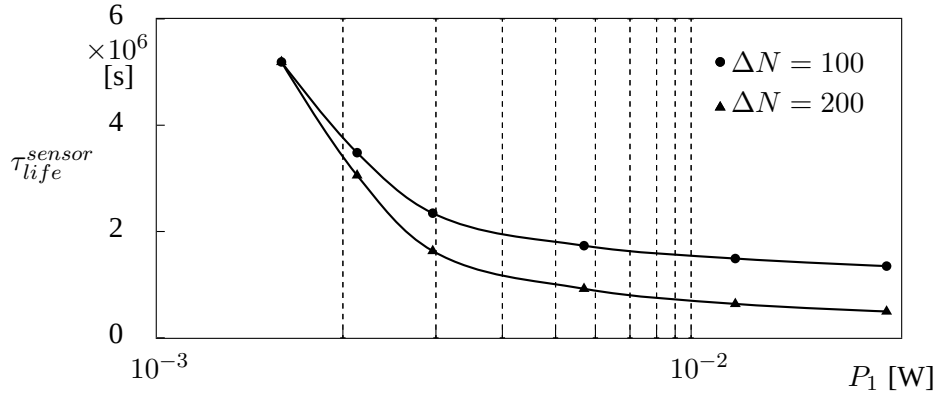



**Fig. 1.** Lifetime of a node as function of transmit power

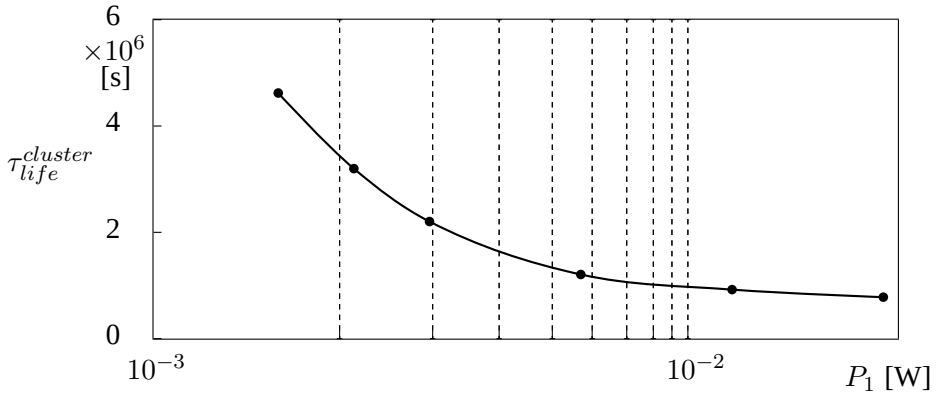



**Fig. 2.** Lifetime of a cluster as function of transmit power

By using the Eq. (13) we obtained the lifetime of an SN as a function of transmit power $P_1$ for a given $\Delta N = 100$, 200 (see Fig. 1). It is evident that the node lifetime decreases as the transmit power increases to support communication in a sparser network.

Considering the lifetime of a clusterhead as a function of the transmit power, the behavior of the lifetime of a clusterhead is plotted in Fig. 2. It can be observed that the lifetime of a clusterhead is less than the lifetime of an ordinary sensor.

The same effect is observed for the lifetime of a hierarchical ad hoc or WSN network (see Fig. 3). This suggests that a long of lifetime is possible when the level of connectivity is small. Thus, the spatial density is low.

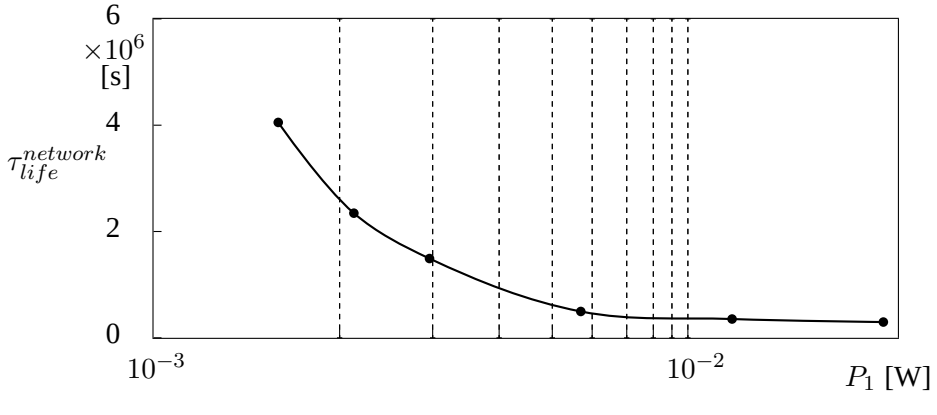



**Fig. 3.** Lifetime of a hierarchical ad hoc/WSN as a function of transmit power

## 5. Conclusion

In this paper we presented a lifetime model for hierarchical sensor and ad hoc networks. This model can be a very powerful analytic tool in WSN design as it can be used to derive many performance parameters of interest. Among other things, it helps showing which organization of an WSN or ad hoc network is better: a flat structure or two-level (multi-level) hierarchy? In general, complete lifetime modeling framework provides an analytic tool for assessing energy consumption in these networks.

Our future work involves using this model as an analysis tool for some routing algorithms in WSNa, as well as damages and mobility in these wireless networks.

## References

[1] C.-K. Toh, "Ad Hoc Mobile Wireless Networks", Prentice-Hall, Upper Saddle River, NJ, 2002.

[2] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, "Mobile Ad Hoc Networking", John Wiley and Sons/IEEE Press, 2004

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks (Elsevier), **38**, 2002, pp. 393-422.

[4] J.-H. Chang, L. Tassiulas, "Maximum Lifetime Routing in Wireless Sensor Networks", in: Advanced Telecommunications and Information Distribution Research Program (ATIRP), College Park, USA, 2000.

[5] I. Kang, R. Poovendran, "Maximizing Static Network Lifetime of Wireless Ad Hoc Networks", in: Proc. IEEE Int. Conf. on Communications, (ICC), **3**, May 2003, pp. 2256-2261.

[6] M. Bhardwaj, A.P. Chandrakasan, "Bounding the Lifetime of Sensor Networks via Optimal Role Assignments", in: Proc. of the IEEE INFOCOM ’02, 2002.

[7] M. Younis, M. Youssef, K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks", in: Proc. 10th IEEE/ACM Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, October 2002.

[8] Q. Li, J. Aslam, D. Rus, "Online Power Aware Routing Wireless Ad-Hoc Networks", in ACM SIGMOBILE, Rome, Italy, 2001.

[9] B. Beferull-Lozano, R. Cristescu, M. Vetterli, "On Network Correlated Data Gathering", in Proc. of the IEEE INFOCOM ’04, 2004.

[10] D. Ganesan, R. Cristescu, B. Beferull-Lozano, "Power-Efficient Sensor Placement and Transmission Structure for Data Gathering Under Distortion Constraints", in ACM Symposium on Information Processing in Sensor Networks (IPSN ’04), Berkeley, California, April 2004.

[11] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Trans. on Wireless Networking, **1**, 4, 2002, pp. 660-670.

[12] B. Chen, K. Jamieson, H. Balakrishnan, R. Morros, "Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", ACM Wireless Networks, **8**, 5, 2002, pp. 481-494.

[13] D.J. Baker, A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm", IEEE Trans. on Communications, **29**, 11, 1981, pp. 1694-1701.

[14] C.V. Ramamoorthy, A. Bhide, J. Shrivastava, "Reliable Clustering Technique for Large, Mobile Packet Radio Networks", Proceedings of the IEEE INFOCOM ’87, 1987, pp. 218-226.

[15] S. Singh, M. Woo, C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", in: Proc. of the 4th ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom ’98) , Dallas, USA, 1998.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 131–140

# Handling of heterogeneous CBR streams in wireless LANs by the Self-synchronised Packet Transfer mechanism

JAROSŁAW ŚLIWIŃSKI [a] ANDRZEJ BĘBEŃ [a] WOJCIECH BURAKOWSKI [a]

[a]Institute of Telecommunications
Warsaw University of Technology
{jsliwins;wojtek;abeben}@tele.pw.edu.pl

**Abstract:** The paper evaluates the effectiveness of the Self-synchronization Packet Transfer (SPT) mechanism for handling heterogeneous Constant Bit Rate (CBR) streams in wireless LAN environment. The SPT mechanism, previously introduced by the authors in [1], is aimed to improve the packet transfer characteristics by limiting packet contentions occurring in IEEE 802.11 MAC protocol. The promising results of our studies obtained in case of homogeneous CBR streams reported in [1], motivate us to extend our investigations for more complex scenario where different types of CBR streams are handled in wireless LANs with erroneous channels. More precisely, we consider heterogeneous CBR streams that differ in both packet sizes or inter-arrival times. For such environment, we propose new algorithm that improves synchronization process of SPT mechanism. The presented results show that thanks to proposed algorithm, the SPT mechanism is able to assure constant transfer delay for almost all transferred packets.

**Keywords:** wireless LANs, CBR streams, Quality of Service, SPT mechanism

## 1. Introduction

Handling of Constant Bit Rate (CBR) traffic in IEEE 802.11 wireless LAN networks [2] with guarantees of low packet transfer delay variation is still not solved in satisfactory way. The main barrier is the contention mechanism used in the Medium Access Control (MAC) protocol. According to it, when two or more stations want to transfer packets after the medium was busy, their transmissions are scheduled randomly following the exponential backoff procedure that with some probability may result in packets collision. Such behaviour of MAC protocol causes that packets transferred in wireless LANs usually experience significant delay variations. This effect was analysed in many papers, let us mention just a few [3], [4] and [5], where authors pointed out delay variation as critical factor for handling delay sensitive traffic over wireless LANs.

In this paper we focus on further evaluation of Self-synchronized Packet Transfer (SPT) mechanism that was originally proposed by us in [1]. Let us recall that the objective of SPT mechanism is to reduce delay variation experienced by packets transferred in wireless LAN.

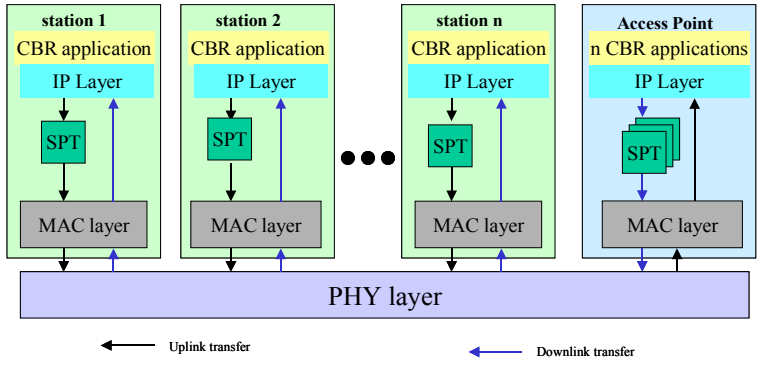



Fig. 1. SPT mechanism in wireless LAN network.

The SPT mechanism synchronizes moments when stations submit packets to MAC layer in such a way to avoid transmission backoffs. Therefore, the SPT mechanism is implemented in each wireless station (including the Access Point) on the top of the MAC layer as presented on Fig. 1. For each CBR stream, the SPT mechanism delays packets for a certain amount of time to assure its submission to MAC layer when the medium is idle. Thanks to this, the packets are transmitted immediately without passing into the backoff procedure. Each SPT entity calculates the value of initial delay independently from others taking into account the time instant when transmission of the previous packet was completed, physical layer parameters, as well as, the packet size.

The excellent behaviour of SPT mechanism in case of homogeneous CBR streams that was reported in [1] motivate us to extend our studies to more complex scenario, where wireless LAN handles heterogeneous CBR streams. In this paper, we analyse CBR streams differing in packet size and packet inter-arrival times that correspond to typical voice codecs. In addition, we study impact of transmission errors occuring in wireless channels on effectiveness of SPT mechanism. For such environment, we propose new algorithm that improves synchronization process of SPT mechanism. This algorithm allows SPT to keep synchronised state in case of sporadic packet retransmissions caused by transmission errors or new call arrivals.

Idea for improved handling of CBR streams is present in many related works. Some of them, like in [8, 9, 10], try to tune the values of MAC parameters. Although it may improve handling and reduce negative effects of packet transfer in wireless LAN, this approach is not able to guarantee strict values of Quality of Service (QoS) objectives required by real-time CBR traffic. On the other hand, this solution is attractive due to the easy implementation in wireless LAN equipment compliant with new IEEE 802.11e standard [14]. The other interesting approach, analysed in [11, 12] and specified as optional feature in [14], extends MAC protocol with the polling mechanism that allows us to emulate synchronous TDMA access. Note that TDMA is the best way for handling CBR traffic, but it requires a centralised

control and complex scheduling algorithm to govern transmission of particular stations. As a consequence, the polling mechanism is seldom implemented in currently available wireless LAN equipment.

The paper is organized as follows. In section 2 we recall details of SPT mechanism, discuss problem of SPT synchronization in case of heterogeneous CBR streams and describe proposed algorithm for keeping synchronised state. Then, in section 3, we focus on evaluation of SPT performances in scenario with heterogeneous CBR streams and erroneous wireless channels. Finally, section 4 summarizes the paper and gives an outline on further works.

## 2. Enhanced SPT mechanism

In this section we recall the SPT mechanism proposed in [1] and then we describe new algorithm that support SPT synchronization process. The objective of SPT mechanism is to reduce delay variation experienced by transmitted packets. This effect is achieved by exploiting the property of MAC protocol, according to which the packets arriving when the medium is idle, must wait DIFS (Distributed Inter Frame Space) time. After that, if the medium is still idle, they are sent without going into the backoff procedure. For that purpose, the SPT running for each CBR stream delays packets for a certain time, called initial delay, in order to submit packet for transmission when medium is idle. The main part of SPT mechanism is the algorithm that determines value of initial delay. This value is calculated as follows. When new CBR connection (with packet inter-arrival time D) arrives at the station, the SPT entity sends first packet immediately to the MAC layer. Then, it observes the time instant when acknowledgement for this packet is received from the MAC layer. For the next packet belonging to this stream, the SPT entity delays the moment when it will be submitted to the MAC layer in order to start its transmission exactly one inter-arrival time (D) after the transmission of the previous packet was started. Formally, the value of initial delay for $n^{th}$ packet may be expressed as (1):

$$d^n_{initial} = T^{n-1}_{conf} - T_t + D - T^n_{arrival} \tag{1}$$

where: $T^{n-1}_{conf}$ denotes time instant when the SPT entity received acknowledgement from the MAC layer about the previous packet $n-1$; $T_t$ is the packet transmission delay that depends on the packet length, MAC overheads and acknowledgement; $D$ is the packet inter-arrival time specific for a given CBR stream and $T^n_{arrival}$ denotes the arrival time of $n^{th}$ packet.

The value of initial delay is calculated independently by each SPT entity during synchronization phase. A given SPT finishes synchronization when the value of initial delay does not change for a few consecutive packets. When all stations finish synchronization, the system works similar to Time Division Multiple Access (TDMA) system where each CBR stream has dedicated slot for packet transmission. As a consequence, it guarantees constant packet transfer delay. The SPT remains in synchronized state until new CBR stream arrives and

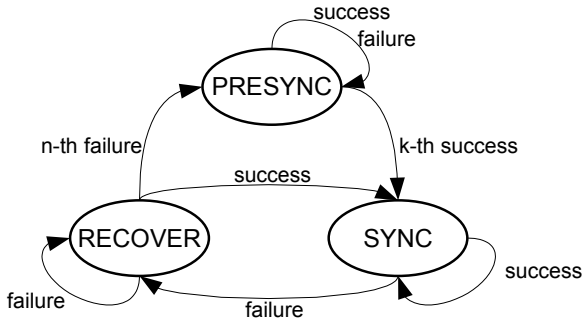



Fig. 2. State machine of enhanced SPT mechanism for handling transmission errors.

causes contention, or until a packet is retransmitted after transmission errors. Such events may cause that some of already synchronized stations may require resynchronization.

The duration of synchronisation phase is one to the key performance factors of SPT mechanism. The results in [1] show that SPT in most cases finishes synchronisation after transmission of a few packets. However, those results correspond to the simplest case of homogeneous CBR streams handled in error free wireless LAN. In this paper, we consider more complex scenario with heterogeneous CBR streams differing in packet size or inter-arrival times handled in wireless LAN with erroneous channels. In this case, SPT synchronization is more difficult as SPT needs to find feasible cycle that takes into account the least common multiple of CBR streams' packet inter-arrival times. The impact of this effect will be studied in details in section 3.

Another problem with SPT synchronisation arises from random MAC retransmission caused by errors over wireless medium or by disturbances introduced by new arriving flow. To cope with them we enhanced SPT mechanism with new algorithm that keeps unchanged value of initial delay in case of temporal disturbances. The algorithm is presented in Fig. 2. Each new flow starts in PRESYNC state and after $k$ successful consecutive transmissions it changes to SYNC state. In state machine description the word *success* means that initial delay of given flow was not changed, and as a consequence, any change of initial delay is marked as *failure*. After achieving SYNC state any single *failure* causes a change to RECOVER state. If in this situation we observe $n$ *failure* events, the state is changed to PRESYNC. Otherwise, any *success* event causes a return to SYNC state. This algorithm improves the behaviour of SPT mechanism in two aspects. It brings stability in case of random packet retransmission resulting from wireless errors and it favours the CBR streams that are already synchronised over the new ones.

Summarising, the most important features of SPT mechanism from exploitation point of view are: (1) stations perform synchronisation process independently in a distributed manner, (2) no changes are required in the MAC layer that can operate according to Distributed Coordination Function (DCF) or Enhanced Distributed Channel Access (EDCA) modes. However, the SPT requires exact values of time: (1) for obtaining the indications from MAC layer

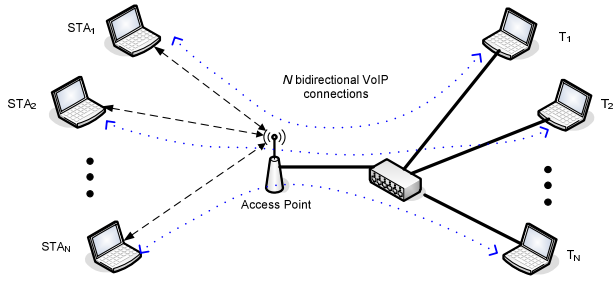



Fig. 3. The wireless LAN network assumed for simulations.

about receiving acknowledge packets, and (2) for emitting packets at specified moments to MAC layer. Let us recall that operating systems, which usually govern logical queues before sending packets to the transmission buffer of wireless card, allow us for controlling the events on the time scale in magnitude of milliseconds. However, our mechanism needs to be more precise as it exploits MAC behaviour scheme, where time intervals are measured in microseconds. Therefore, even if the MAC layer is unchanged, some support from the low level software, called firmware, may be needed.

## 3. Performance evaluation

In this section we show the performance of SPT mechanism assuming different CBR streams. In the first part, we study the effectiveness of SPT mechanism for handling heterogeneous CBR streams. Next we evaluate the impact of erroneous wireless channel on SPT. For obtaining numerical results, we used NS-2 simulator [13] enhanced with the model of the SPT mechanism.

For all experiments we assume a simple wireless LAN network depicted on Fig. 3. The network consists of a single access point (AP), a number of wireless stations ($\mathrm{STA}_x$) and a number of wired terminals $\mathrm{T}_x$. The wireless LAN operates in DCF access method using 11 Mbps physical layer with long preamble. Following results from [1], we configure the value of minimum contention window ($\mathrm{CW}_{min}$) to 4, while maximum contention window ($\mathrm{CW}_{max}$) is equal to 1024. Furthermore, we assume that all STAs and AP are enhanced with the SPT mechanism. Additionally, the wired part of the system is over-provisioned and, as a consequence, has no impact on the collected packet transfer characteristics.

Three types of bidirectional CBR connections may run in the system that are presented in Tab. 1. Connections of type A and C corresponds to G.729 and G723 voice codecs, while type B was fixed arbitrary to emit long packets every 20ms. In each tests, connections are started randomly between pairs of wireless station $STA_x$ and corresponding wired terminal $T_x$.

| Parameter | Profile A | Profile B | Profile C |
|---|---|---|---|
| Transport protocol | UDP | UDP | UDP |
| Connection type | bidirectional | bidirectional | bidirectional |
| Inter-arrival time (ms) | 20 | 20 | 30 |
| Packet size (bytes) | 60 | 1500 | 64 |
| IP peak bit rate (kbps) | 24 | 600 | 17 |

Table 1. Traffic profiles for connections used in simulations.

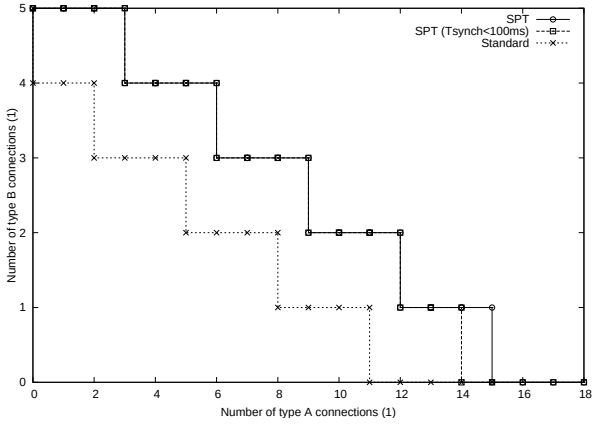



Fig. 4. System stability region for mixture of type A and type B connections.

### 3.1. Heterogeneous CBR connections

In the first experiment we focus on stability of system handling connections that share the same packet inter-arrival period, but differ in packet sizes. For this purpose we choose type A and type B connections and configure physical layer to not introduce any errors. Results of simulations are presented in Fig. 4 in a form of stability regions for three cases: (1) standard system, (2) SPT enhanced system with synchronisation time limited to 100 ms, and (3) SPT enhanced system without a limit on synchronisation time.

In all cases SPT exceeds or equals the capacity of standard system. When only type A connections are considered, we can handle 18 connections instead of 15. When only type B connections are handled, we do not observe such capacity gain. This is an effect of low number terminals contenting for access to the wireless medium and SPT cannot provide improvement in this case. We also note that in almost all cases achieving synchronisation is very fast and takes less than 100 ms. We conclude that when all connections share same packet inter-arrival period, then they can be handled effectively by SPT mechanism.

Next experiment involves connections that have similar packets sizes and differ in packet inter-arrival period. Therefore, simulations cover here type A and type C connections. Fig. 5 shows system stability regions for this scenario. The curves represent the same meaning as

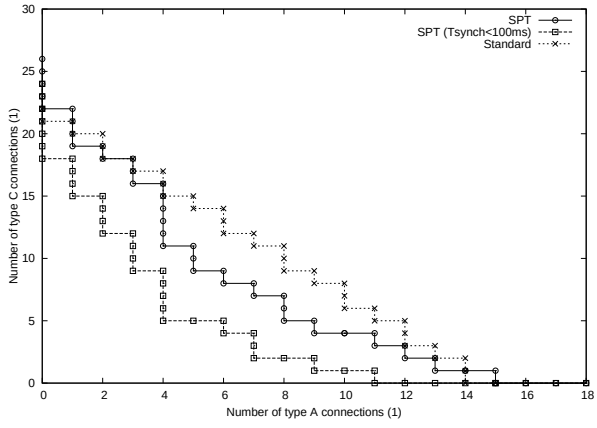



Fig. 5. System stability region for mixture of type A and type C connections.

| Connection profile | FER=0 | FER=$10^{-4}$ | FER=$10^{-3}$ | FER=$10^{-2}$ |
|---|---|---|---|---|
| A | 18 | 15 | 15 | 13 |
| C | 26 | 23 | 22 | 18 |

Table 2. Capacity of wireless LAN with SPT mechanism for different FER values.

in previous experiment.

We see that standard system is more versatile when a mixture of type A and type C connections is handled. Acceptance region behaves in almost linear way. SPT tends to slightly lower values when the number of connections with different characteristic is low (few at most). However, when we impose limitation on synchronisation time, then the capacity in middle range of graph is much lower. When we consider homogeneous cases, where there is zero connections of one type, then capacity and synchronisation time improves. We conclude that connections with different packet inter-arrival periods should be avoided in SPT mechanism and we propose to allow only one global value that is used for handling connections by SPT.

Following above conclusions, we compared the values of IPDV for points selected from Fig. 4 for acceptance region of standard system. As we see on Fig. 6, the IPDV in standard wireless LAN system increases with a growing number of type B connections in the system. This effect originates from the combination of traffic with different packet sizes and is enhanced by random behaviour of MAC protocol. Nevertheless, when we use the SPT mechanism this problem is mitigated and every flow in the synchronized system observes IPDV equal to zero.

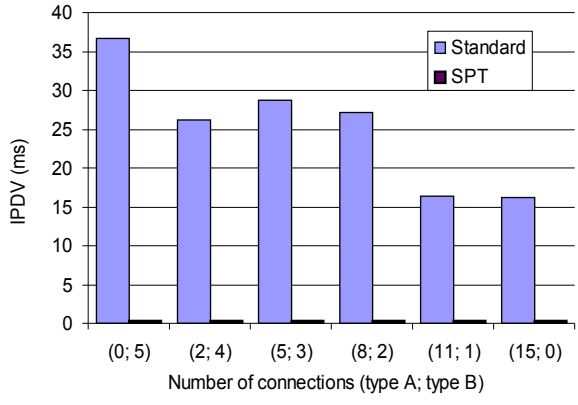



Fig. 6: Comparison of IPDV values of standard and SPT enhanced wireless LAN obtained for limit points of standard system from Fig. 4

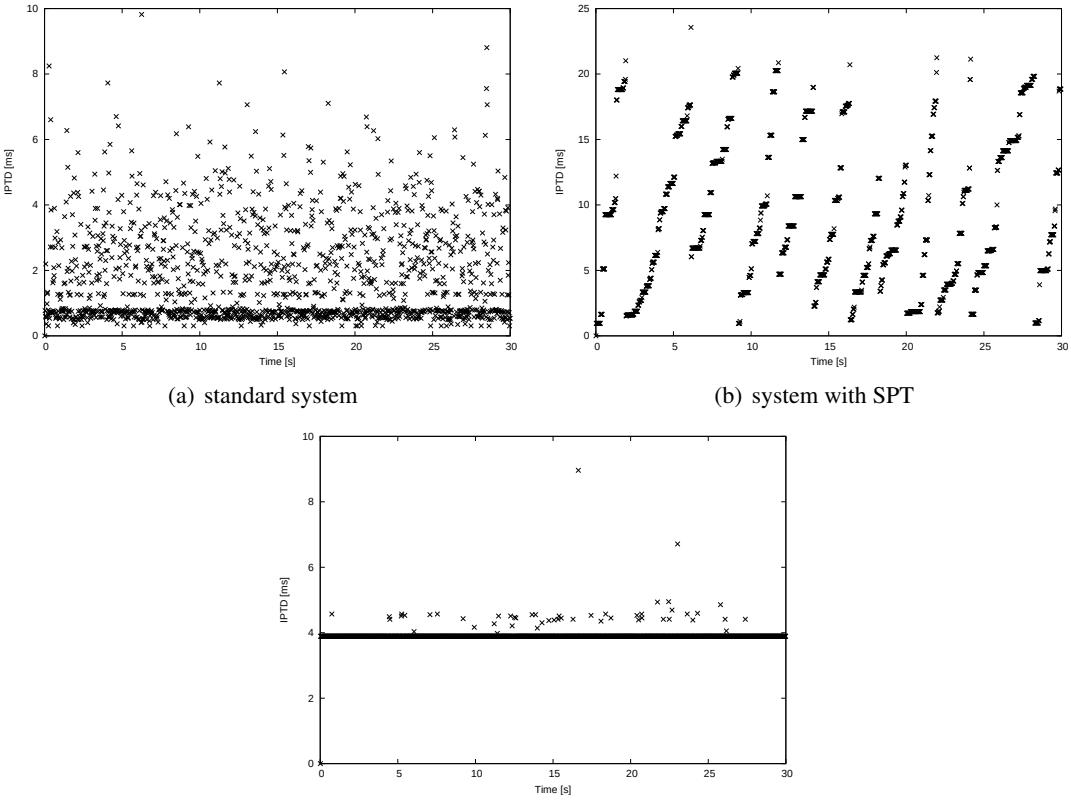
### 3.2. Erroneous wireless channel

Second part of evaluation covers the behaviour of SPT in the presence of random independent transmission errors. As we do not modify MAC protocol behaviour, every transmission error of data frame or acknowledgement frame will cause retransmission and thus, it changes the delay that packet spends in MAC layer. Tab. 2 provides simulated SPT system capacity for different values of Frame Error Rate (FER), where state machine parameters are $k = n = 5$. Although for low values of FER ($10^{-4}$ and $10^{-3}$) the SPT mechanism suffers from random retransmissions, it is able to synchronise for the same admissible load as standard system. In case of moderate FER=$10^{-2}$ the SPT is not able to cope with high level of randomness in system and requires lower system utilization. Lower load increases distances between packet transmissions of different streams and it allows for control of jitter caused by transmission errors.

Fig. 7 show time plots of IP Packet Transfer Delay (IPTD) metric observed by single flow in system under moderate FER=$10^{-2}$: (1) without SPT, (2) with SPT, and (3) with enhanced SPT. In assumed traffic scenario the system handles only one type of connections. The number of connections is chosen from Tab. 2 for type A. For the case on Fig. we see 7(b) frequent resynchronisations that form increasing patterns. However, after enhancing SPT we mitigate random retransmissions and achieve almost constant value of IPTD, as on Fig. 7(c).

## 4. Conclusions

In this paper we presented evaluation of SPT mechanism for the case when heterogeneous CBR streams are considered. Additionally, we enhanced the SPT by introducing state machine for improving synchronisation stability in error prone wireless environment. Sim-




(a) standard system

(b) system with SPT

(c) system with enhanced SPT

Fig. 7. IPTD for single packet stream in system with FER=$10^{-2}$ and 13 connections of type A.

ulation experiments confirm that for low level of FER the SPT is able to achieve the same capacity as standard wireless LAN. When transmission errors became frequent, the SPT is still able to synchronise for lower system load and it provides constant packet transfer delay for almost all transferred packets. As SPT is very similar to shaper mechanism we may also expect that it should allow for operation with bursty traffic sources.

## References

[1] J. Sliwinski, W. Burakowski, A. Beben: A Method for Improving Transfer Quality of CBR Streams over Wireless LANs, *In Proc. of 4th Polish-German Teletraffic Symposium*, 2006, September 21-22, Wroclaw, Poland, ISBN 83-7085-975-5

[2] IEEE 802.11 WG: Part 11: Wireless LAN MAC and physical layer specifications, *IEEE Standard*, 1999

[3] D.P. Hole, F.A. Tobagi: Capacity of an IEEE 802.11b wireless LAN supporting VoIP, *Proceedings of IEEE International Conference on Communication*, ICC 2004

[4] S. Garg, M. Kappes: Can I add a VoIP call?, *Proceedings of IEEE International Conference on Communication*, ICC 2003, Alaska

[5] K. Medepalli, P. Gopalakrishnan, D. Famolari, T. Kodama: Voice capacity of IEEE 802.11b, 802.11a and 802.11g wireless LANs. *In proc. of IEEE Global Telecommunications Conference*, Globecom 04, Dallas, USA 2004

[6] F. Anjum, et al: Voice performance in WLAN Networks - an experimental study, *In proc. of IEEE Global Telecommunications Conference*, Globecom 03, San Francisco, USA 2003.

[7] ITU-T Recommendation: Y.1541, Network performance objectives for IP-based services, ITU, May 2002

[8] A. Banchs, X. Perez: Providing Throughput Guarantees in IEEE 802.11 Wireless LAN, *IEEE Wireless Conference on Networking Communication (WCNC 2002)*, 2002

[9] A. Veres, A.T. Campbell, M. Barry, Li-Hsiang Sun: Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control, *IEEE Journal on selected Areas in Communications*, vol. 19, pp. 2081–2093

[10] A. Jain, D. Qiao, K.G. Shin: RT-WLAN: A Soft Real-Time Extension to the ORiNOCO Linux Device Driver, *2003 International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2003)*, 2003

[11] M. Veeraraghavan, N. Cocker,T. Moors: Support of voice services in IEEE 802.11 wireless LANs, *In Proceedings of the IEEE INFOCOM 2001*, 2001

[12] Qiang Ni: Performance Analyzis and Enhancements for IEEE 802.11e Wireless Networks, *IEEE Network*, July/August 2005

[13] NS-2: Network Simulator 2. available on http://www.isi.edu/nsnam/ns/

[14] IEEE 802.11 WG: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications; Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, *IEEE Standard*, 2003

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 141 - 150

# Cross-platform solution for telecommunication networks simulation and management. Version for pocket PCs and smartphones.

IVANNA DRONYUK [a] ROMAN KOSHULINSKY [a]

[a] Institute of Computer Sciences and Information Technologies
Lviv Polytechnic National University
*ivanna.droniuk@gmail.com*

**Abstract:** A cross-platform solution developed for telecommunication networks simulation and management. Structure of the application is described thoroughly. Optimization possibilities of program are demonstrated on example and analyzed. Its perspectives and spheres of use are mentioned.

**Keywords:** telecommunication network, algorithm on graph, traffic optimization, object-oriented programming.

## 1. Introduction

In modern period of total informatization a rapid development of wide variety of networks (including telecommunication - TCN) leads out of the network functioning optimization problem to the rank of extraordinary actual and practically important problems [1]. Modern information technologies open new possibilities for simulation and different optimization algorithms implementation. Nowadays together with desktop PCs notebooks and pocket PCs and smartphones are widely used. So it's reasonable to support application which performs simulation and optimization of telecommunication networks for mobile platforms. This allows solving problems of optimal network management.

Program for telecommunication networks simulation and optimization was developed using language C# oriented on platform .NET. The tool used for development was Microsoft Visual Studio 2005. Such choice was made because C# language is object-oriented (that essentially simplifies developing large projects), safe, powerful and satisfies all engineering requirements. Using object-oriented method of attack an interface of abstract types was used [2]. Creating an abstract data model is one of the most important elements in solving problems using computer. For telecommunication network visualization own component (also called "control" in [3]) was created. For optimal paths calculations own algorithms were implemented. They are based on classical minimal path and maximal flow calculation algorithms.

## 2. Mathematical model of telecommunication network.

Telecommunication network (TCN) can be considered as an aggregate of nodes (commutators) and an aggregate of connections between pairs of such nodes. Let telecommunication network contain N nodes. All given nodes can be considered as some set containing N elements. Let's mark this set as V. Also let's mark as $V^{(2)}$ a set of all 2-element subsets in set V. It's obvious that any subset E of set $V^{(2)}$ ($E \subseteq V^{(2)}$) will represent some aggregate of connections between nodes. Let's submit for consideration pair G = {V, E} which is called graph. Set V represents a set of graph's vertexes and E – a set of graph's edges. Set of vertexes V in graph simulates nodes in TCN and set of edges E – connections between these nodes.

Graph G is called directed when $G=\{V,E\}$ where $E \subset V \times V$ is a subset of Cartesian square of set V. Edges of directed graph are ordered pairs $(s,t) \in V \times V$. Graph G is called planar when it could lay on the plane so that edges would't have any other points of intersection except graph's vertexes. TCN in most cases is planar so it can be placed on 2D plane. Graph that simulates such network will be planar. This feature allows representation of network without edge intersections.

A sequence of edges $(l_1,\ldots,l_n)$ where every two consecutive edges $l_i$, $l_{i+1}$ are different and have common vertex is called path in graph. The same edge could be met several times in one path. Vertex of edge $l_1$ which doesn't belong to $l_2$ is called beginning of path and vertex of edge $l_n$ which doesn't belong to $l_{n-1}$ is called end of path. Two vertexes s and t are called connected when there is a path with beginning in s and ending in t. When vertexes s and t are connected and s = t then path is called cyclic. When all vertexes in graph G are connected with each other then graph is called connected. It's obvious that problem of graph connectivity is extremely important for TCN because connectivity breach in TCN means impossibility of connection establishment between subscribers that is inadmissible.

Main characteristics of each graph are its vertexes and edges quantity, N and K correspondingly. These characteristics represent quantity of nodes in TCN and connections between them. When $l \in E$ looks like $l=\{s,t\}$ then edge is called incident on vertex s and vertexes s, t are called incident on edge l. Vertexes s and t that belong to the same edge are called adjacent. Quantity of edges that are incident on vertex s is called vertex's degree and marked as degree(s). Degree of vertex determines importance of appropriate node in TCN. Depending on type of simulated network some constant $C_m$ which determines importance of node in network could be used.

When the following condition is met:

$$degree(s) \geq C_m \qquad (1)$$

or in other words when degree of vertex is greater than some constant value then current node could be considered as central, in other case – as peripheral. Vertex s is called final when its degree is equal to 1: degree(s)=1.

For numeric realization of optimization algorithms let's introduce concept of graph density. Let's submit for consideration $\eta$ - density of graph. It's calculated with the following formula:

$$\eta = r/N \qquad (2)$$

Graph is dense when $\eta \geq N$ and sparse in opposite case. Most of telecommunication networks usually use only small part of all possible connections. For example, graph containing main lines of TCN "Ukrtelecom" consisted of 42 nodes (N=42) and 72 edges (K=72) in 2004. Maximal quantity of edges for such graph is 861 ($Z_{max}$=861) but really it contains only 72 edges ($\eta$=3, 4). It means that graphs of TCN are usually sparse. This feature of TCN graphs is extremely important while choosing optimization algorithms because some of them perform much faster on sparse graphs [2].

Graph G is called edge-weighted when with each edge l positive real number $w$(l) is associated, or in other words there is a reflection $w$ from set of edges E to set of real numbers $w$:E→R. Number $w$(l) is called weight of edge l. Sum of all edge weights is called weight of graph edges and marked as $w$(G).

During TCN graph consideration there is also a need to take some characteristics of nodes into account. So it's reasonable to submit weights of graph nodes for consideration. Graph G is called vertex-weighted when with every its vertex s a positive real number $\omega(s)$ is associated, or in other words when reflection $\omega$ from set V to the set of real numbers $\omega$:V→R is defined. Number $\omega$(s) is called weight of vertex s. Sum of all vertexes' weights is called weight of graph vertexes and marked as $\omega$(G).

For TCN simulation weighted graphs are usually used. Such consideration allows taking some features of connections' structure into account. For TCN the following kinds of edge weight reflection are important:

$$w(l)=d(s_i, s_j),\ d:E\rightarrow R, \qquad (3)$$

- reflection that defines distance between nodes $s_i$, $s_j$;

$$w(l)=t(s_i, s_j),\ t:E\rightarrow R, \qquad (4)$$

- reflection that defines signal passing time between nodes $s_i$, $s_j$;

and also

$$w(l)=c(s_i, s_j),\ c:E\rightarrow R, \qquad (5)$$

- reflection that defines carrying capacity of connection between nodes $s_i$, $s_j$.

In a similar way let's submit for consideration reflection of each vertex's weight in graph:

$$\omega(s)=\theta(s_i),\ \theta:V\rightarrow R \qquad (6)$$

- reflection that defines signal delay in node $s_i$,; and also

$$\omega(s)=h(s_i),\ h:V\rightarrow R. \qquad (7)$$

During optimization algorithms implementation in computer system a possibility of data inputing and storing for each of reflections (3)-(7) was provided. Reflection (6) which describes signal delay in node is represented as

$\omega(s)=0$ when there is no delay on node, (8)

$\omega(s)=\theta_1$ - delay that persists on node s in overloaded mode,

$\omega(s)=\theta_2$ - delay that persists on node s in normal mode,

$\omega(s)=\theta_3$ - delay that persists on node s in idle mode.

Reflection (7) that describes different working modes could be represented as following:

h(s)=0 when node s is in normal mode, (9)

h(s)=1 when node s is in overloaded mode,

h(s)=2 when node s is in idle mode,

h(s)=-1 when node s is disabled.

Considered characteristics of TCN are taken into account in developed computer system. So it's easy to represent importance of current node in network, various working modes of TCN and operate on different parameters such as delay time, distance between nodes, signal passing time, node's availability.

## 3. Structure and general characteristics of project

Project Graph .NET is oriented on desktop and mobile platforms and runs under Microsoft .NET Framework (.NET Compact Framework). It allows simulating telecommunication networks using graphs theory and solving flow optimization and management problems. As mentioned earlier, nowadays smartphones and pocket PCs are widely used. That's why project was developed taking mobile platforms into account.

Solution Graph .NET is cross-platform: in case of desktop PCs and notebooks it works under operating systems Microsoft Windows 2000/XP/Vista after installing .NET Framework 2.0 or higher, in case of pocket PCs and smartphones this is Windows Mobile 2003/5.0/6.0 with .NET Compact Framework 2.0 or higher. It's important to note that for correct work on smartphones a sensor display is required because most of operations on graphs (add/remove nodes, relations, etc.) couldn't be performed using only keyboard. Target platforms and operational systems which support current project are represented on fig. 1. Version that is under consideration marked with grey color.

Such choice was made because developed application is multi-purpose and allows simulation of other kinds of networks, for example transportation ones. Then nodes will represent cities, crossroads and some travel points inside cities or railway stations. Connections in this case will represent roads or railways. This opens new possibilities for program in route planning. In this case developed computer system could be a part of GPS navigators and other route planners which are mostly oriented on mobile platforms.

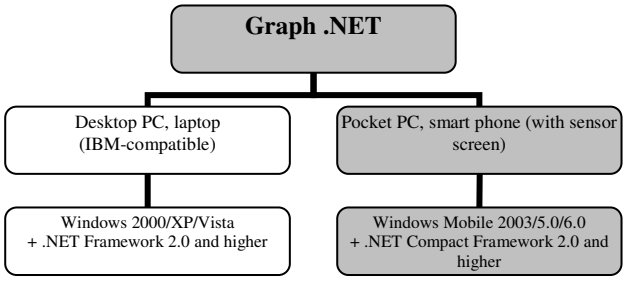



Fig.1. Target platforms and operational systems supported by project Graph .NET.

Project consists of two parts which are compiled separately. First one is core which includes all base classes for representing and processing graphs and is compiled into library Core.dll. It contains implementations of algorithms on graphs including networks features, component for graph visualization and classes for graphs serialization into XML files. Second part contains user interface and is compiled to executable file Graph.net.exe. The advantage of such approach is that algorithms and user interface are separated so that others can write their own user interfaces or modify existing one without investigating algorithms implementation and vice versa. It's important to note that .NET Framework and .NET Compact Framework are slightly different so for compatibility purposes core for mobile platforms a little differs from core for desktop computers. Structure of project is represented on fig. 2.

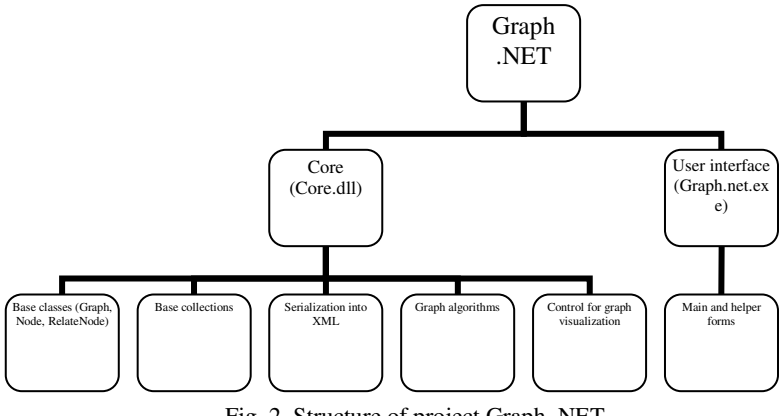



Fig. 2. Structure of project Graph .NET.

Developed application has user friendly interface. Main window contains main menu 1, component for graph visualization and management 2 and tabs with some additional settings 3 (Fig. 3).

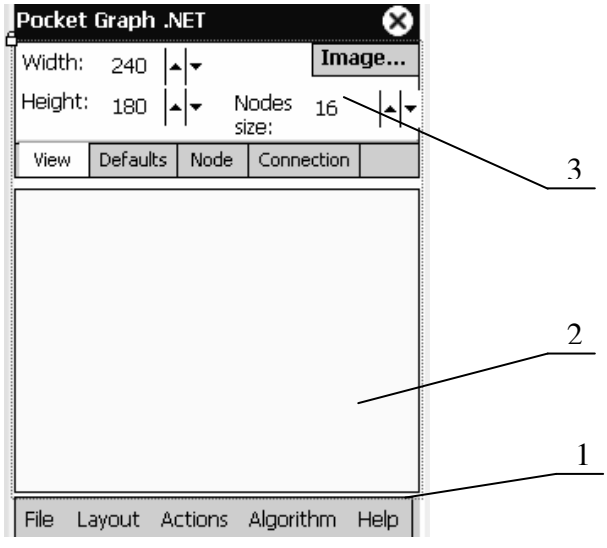



Fig. 3. Main window of program GRAPH.NET (version for mobile platforms).
1 – main menu, 2 – component for graph visualization and management
3 – tabs with additional settings

Result of calculations is displayed as a message. In case of shortest path algorithm it is a value of calculated path length. When maximal flow algorithm is selected then value of maximal flow is displayed. Unit of measurement depends on initial values.

During program development the following base classes were created – Graph (represents graph), Node (represents vertex of graph), Connection (represents edge).

Vertexes in graph (or nodes in telecommunication network) have following properties: *Delay* – delay of signal on node; *Mode* – working conditions of selected node; *Size* – screen size of selected node (in pixels); *Text* – contains text which is displayed over node.

Edges in graph (or connections in telecommunication networks) have following properties: *Capacity* – capacity of selected connection; *Distance* – length of the selected connection; *Enabled* – determines whether selected connection is available at the moment; *Flow* – shows value of flow which is passing through selected channel at the moment.

The most compatible mode for developed application is QVGA, vertical screen orientation (240x320). VGA mode (480x640) is also supported, but layout of all main elements is optimized for QVGA display. When screen resolution is not standard (i.e. 800x480, 320x320, etc.) graph visualization component can be adapted to needed sizes using additional settings on tabs in main widow. Also it's important to note that if size of graph is larger than screen resolution, scroll bars appear. They allow gaining access to whole graph despite small screen size.

Graph is a hierarchical structure so in addition to base classes were created classes that contain collections of nodes and edges. All data about graph could be saved to and

loaded from XML files. In program there is a feature which allows loading images of geographical maps and setting them for graph as “wallpapers”. It is extremely useful when program is used as electronic map and nodes in graph represent towns and cities. The advantage of XML format is that it is “human-readable” and user can edit graph manually, using text editors.

Sometimes in networks from different reasons nodes or connections between them could be disabled. In that moment all flows should be redistributed using available nodes and connections. In application all these conditions are taken into account. Also network can work in different modes (idle, normal, overloaded). In our model this is also taken into account and it determines delays on nodes.

## 4. Optimization possibilities of program

For TCN traffic optimization purposes algorithms of shortest path and maximal flow were developed. As it was mentioned earlier, developed application allows calculating minimal paths between two nodes. Shortest path algorithms are based on known classical Dijkstra and Floyd algorithms [2] but some modifications were made taking features of TCN reflected in formulas (1)-(7) into consideration. In classical Dijkstra algorithm minimal path is calculated by building shortest path tree (SPT), nodes to which are added in order of increasing distance to them from initial node. In Floyd algorithm value is received from the appropriate matrix. Classical Dijkstra and Floyd algorithms are widely described in [2]. Modified in developed computer system shortest path algorithms take delays on nodes defined by reflection (6) into account.

In case of telecommunication networks we could find the shortest connection between some nodes. From the other side, this possibility could be used in GPS navigation for building optimal paths between cities (cities in this case are represented by vertexes of graph) or inside some city (vertexes of graph in this case should represent parts of streets).

Flow calculations are based on classical augmental way algorithm, known in literature as Ford-Fulkerson algorithm. It allows finding maximal flow that could be passed between current two nodes using all available at the moment connections. Developed computer system calculates value of maximal flow and shows its structure by highlighting appropriate connections. Running application on pocket PC calculating maximal flow in city telecommunication network is shown on fig. 4.


Fig. 4. Program Graph .NET on pocket PC. Calculating maximal flow inside city telecommunication network.

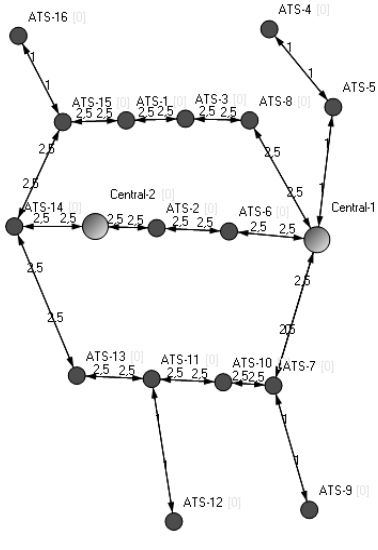



Fig. 5. Structure of simulated city telecommunication network. Screenshot from Graph .NET for desktop PCs.

Now let's demonstrate possibilities of program on some example. Telecommunication network shown on fig. 5 consists of 2 central and 16 peripheral nodes. All central and 11 peripheral nodes form two circuits; other 5 nodes lie beyond them. Carrying capacity of conections inside circuits is equal to 2,5 Gbit/s, beyond them – 1 Gbit/s. In every moment of time some connections could be disabled. Delays on all nodes for simplicity are equal to 1 ms. Speed of signal passing is equal to the speed of light so delays on conections inside city could be ignored. Table 1 shows calculated maximal possible flow, shortest path and passing time for signal in different situations.Central nodes ("Central") are marked with letter C, peripheral ("ATS") – P.

First row in table shows situation when both nodes are situated inside circuits so maximal flow and minimal signal passing values received. Second and third rows show situation when one of connections which belong to previous shortest path is getting disabled. Program recalculates shortest path looking for available nodes but this increases signal passing time. Maximal possible flow decreases because quantity of possible paths between nodes has been decreased. Next two rows show the similar situation but in this case peripheral connections are bottleneck of the entire path so maximal possible flow is equal to carrying capacity of peripheral connections. Last row in table shows situation when peripheral connection is getting disabled. In this case program is unable to find alternative path and appropriate message is shown.

| Source node | Type of nodes in accordance with formula (1) | Desti-nation node | Disabled connections | **Signal passing time, ms** | Shortest path of signal | **Maximal possible flow, Gbit/s** |
|---|---|---|---|---|---|---|
| C1 | Central | C2 | - | **2,0** | C1-P6-P2-C2 | **7,5** |
| C1 | Central | C2 | P6 – P2 | **6,0** | C1-P8-P3-P1-P15-P14-C2 | **5,0** |
| C1 | Central | C2 | P6 – P2<br>P3 – P1 | **6,0** | C1-P7-P10-P11-P13-P14-C2 | **2,5** |
| P4 | Peripheral | P9 | P6 – P2 | **3,0** | P4-P5-C1-P7-P9 | **1,0** |
| P4 | Peripheral | P9 | C1 - P7<br>P13 – P11 | **10,0** | P4-P5-C1-P6-P2-C2-P14-P13-P11-P10-P7-P9 | **1,0** |
| P4 | Peripheral | P9 | P5 – C1 | **-** | - | **-** |

Tab. 1. Calculating maximal flow and shortest path of signal in city telecommunication network.

Calculation time wasn't considered in this table because it is less than minimal detectable by program value (0,1 ms). Initialization time was equal to 15 ms.

## 5. Conclusions

1. Created computer system for telecommunication networks simulation. A cross-platform application developed using language C# and object-oriented point of view.
2. Theoretical base for computer system development is presented.
3. Program has user-friendly interface and allows solving optimization problems for different kinds of networks. In case of telecommunication network nodes are represented by vertexes in graph and connections by edges.
4. Maximal flow and shortest path calculation results for main lines in city telecommunication network were presented on example and analyzed.
5. In real networks from different reasons some connections or nodes could be disabled. In developed application this feature was also taken into account and transportation flows could be redistributed.
6. Project is in stage of further development. Versions for desktop and mobile systems are improving simultaneously. Perspective branch of development is adapting project for GPS navigators and other route planners.

# References

[1] *Base Mathematical Theory of Telecommunication systems* (edit by V.V.Popovsky).-Kharkiv-"SMIT company", 2006.-P.564. (in Ukrainian).
[2] Sedgewick R. *Algorithms in C++.Part 5.Graph algorithms.* Princeton University-Diasoft,-Kiev: 2002.- 496pp.(in Russian).
[3] J. Richter. *Applied Microsoft .NET Framework Programming.* Microsoft Press, 2002, 556 pp.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 151–164

# Traffic engineering for industrial networks

Michał Morawski [a]

[a]Division of Computer Networks
Technical University of Łódź
morawski@zsk.p.lodz.pl

**Abstract:**

In the paper we present the results of our further work on the new method of the traffic engineering based on an adaptive multipath unidirectional routing based on the *Minimum Delay Routing* principle [10,11,30–33]. The routing problem considered in our work is focused on the traffic specific for industrial applications and low performance links, esp. wireless. In such situation the regular on-off low volume traffic is interlaced with the intensive stream and/or datagram traffic. The traffic specific for control of technological processes requires quite low bandwidth, reconciles with even large single (not clustered) data loss, but does not tolerate delays. These cause significantly different requirements than in typical networks. The paper extends previous results considering TCP traffic by maintaining paths and overall network stability in hard traffic conditions.

In the presented approach, we assume that the values of link costs in all links and all metrics are not known exactly, but we consider them as values with uncertainty. Such an approach, together with associated forwarding method allows to assimilate well known routing algorithms (typically different for wired and wireless parts of networks) to the behaviour close to optimal, and therefore, to obtain significantly shorter latencies, jitter, nearly no loses, better throughput for data flows, than in the case of usage pure standard or uniform algorithms.

The paper strictly extends the work published in [24].

**Keywords:** : Traffic Engineering, Routing, QoS, Industrial networks

## 1. Statement of the problem of the traffic engineering in industrial networks

The industrial networks, i.e. networks, that connect actuators, sensors and control nodes necessary to control technological process are traditionally divided into three classes: the fieldbus, real-time and backbone [19], where fieldbus networks are usually developed using RS-485 controlled devices under supervision of simple master-slave protocol like e.g. MODBUS, however there are plenty of incompatible “standards” in this kind of networks. Such devices periodically interchange information, that consists of a few bytes only, but sometimes the measurements produce the large volume data (i.e. cameras and scanners). The number of such devices is in range of thousands or tens of thousands for large plants.

Due to the channel communication limitations, often lack of collision sensing, the master-slave or command-answer or "external" policy is preferred (esp. on low throughput links) over autonomous (time of the transmissions is chosen by the device). Sometimes the event driven schedule is applied [19]. For large latency links and clustered informations it is an equivalent to the stream transport. In practice both policies time or event driven are used simultaneously.

The "real-time" network is a time predictable network like Profibus [21], Real-Time Ethernet [1], ARINC family [6, 19], etc., that connects processing nodes responsible for collecting data for the controlling processes using fieldbus. The backbone network is a general purpose network without any special requirements.

Today however, the border between a fieldbus and a real-time network wipes out. Most devices use OPC [2] standard of a communication, or very similar, but highly simplified paradigm, if no enough throughput or processing power is available. OPC-aware devices are extremely easy to deploy, but require rather high bandwidth, and are better adjusted to high speed, low latency, lossless wire links than to the more liable and slow wireless or asynchronous ones. Moreover OPC require a stream (point-to-point) transport and therefore there is a problem with redundancy at this level. The afore-mentioned simplifications are to adjust fieldbus control nodes to the OPC.

Moreover the traditional fieldbuses suffer from the poverty of scalability and redundancy. Increasing the number of sensors/actuators sometimes is very expensive due to depletion of channel resources (distance, power, throughput, etc.). Redundancy required for safety of technological processes associates not only with a redundancy of sensors – redundancy of links is also required. All the channels must fulfil the time constraints, not only during normal work, where periodic communication usually dominates, but even particularly in an alarm or emergency mode work, where multiple of events are transmitted in a burst mode. Violating of these can cause a catastrophic situation like the famous failure of F18 prototype in 1993 [19].

Therefore the wireless links more and more often complement the wired ones, giving additional advantages by decreasing weight of systems (e.g. aircrafts) [26, 29], making available to control mobile devices, decreasing costs of installation and maintenance, etc.

Such approach caused a development of the special kind of networks like ad-hoc [25], or especially sensor networks [9]. The main development efforts in this domain are directed to military or environmental research, but industrial standards were also created – e.g. IEEE 1451 [3].

For such networks there is necessary to develope the common for wired and wireless parts traffic engineering algorithms, that can direct, or even split the traffic into multiple channels.

It is necessary to remember, that although IP protocol dominates in all traditional network, the kind of networks described above accepts it very slowly due to the significant overhead and bandwidth limitations. However such solutions like headers compressions [15], tunnelling, gateways can be easily and willingly applied. Independent of this, one

can consider UDP (less or more CBR for autonomous communication) traffic and TCP (for event driven communication or asynchronous sinks [2]).

The Traffic Engineering (TE) applied to the MPLS networks has significantly different requirements, therefore different algorithms are applied to these [16]. Esp. no contracts are defined for the technological processes, and it is impossible to predict all possible alarms, conditions or faults that can influence on the traffic volumes.

## 2. Optimal usage of links

About thirty years ago, when networking and its theory was emerging, Gallager [10] formulated the *Minimum Delay Routing* principle of the optimal routing. After appearing of the predecessor of today Internet – the ARPANET, many attempts of the practical application of this principle have been made [17], but without success. The main problem with such an application was the arising of routes flapping, e.g. oscillations of routes (paths) caused by changing the paths delays, that is a result of changing the link loads. The flapping phenomenon influences unfavourably throughput of the network as a whole, and by least increasing the network latency and decreasing quality of the network device control.

The adaptive routing is considered as an unstable solution and therefore, the idea of optimal routes was dropped twenty years ago [17], and metrics used today are based on the simple approximations of the optimum, which are administratively set up and hence constant. These approximations are usually based on hop counts, or link attributes like available bandwidth, reliability, average delays, loss ratio and others. When a network is stable (in the sense of the carrying the stable traffic), every such an attribute set properly (or composition of the attributes) better or worse approximates the optimum. However, the assumption of network stability (the link loads are constant) is unrealistic. Therefore such a typical approach causes significantly less then optimal network usage, and less efficiency, i.e. growing costs of networks and can be dangerous for the control process leading even to a diaster.

The main goal of our work is the maximum exploitation of existing (very limited in fieldbus) network resources, taking into account the properties of the standard transport and application protocols without necessity of uniforming them in the whole network (routing protocols in wire and wireless infrastructure, ad-hoc, sensor parts of the network significantly differ). The method of achieving this goal, has to cooperate and complement Active Queue Management (AQM) algorithms [12, 28], although the AQM has significance only for the event-driven mode or when sensors have highly different periods, and queues lengths have to be kept short to maintain low latencies.

Admittedly the notion of a routing protocol is inseparably connected to a way of finding routes in the network, however the proposed algorithm is **not** a routing algorithm in such meaning. It is rather a method of the link cost manipulation in order to increase efficiency of any standard routing protocols. The advantages of the proposed algorithm can be observed only when multiple routes that satisfy LFI (*Loop Free Invariant*) condition,

between the same pair of the nodes existing in the network. This approach can be applied both to connection (in particular – in the highly cultivated today MPLS networks [16, 27]) and connectionless networks, if the route establishing algorithm supports multiple paths. Note, that the problem of establishing paths that satisfy the LFI condition is not a topic of this paper (and our work), and it is exhaustively described in [7].

The proposed algorithm is suitable for the unicast flows, but it is necessary to remember, that the multicast (or better – anycast) flows can add additional redundancy.

## 3. Multipath routing

The idea of multipath routing has been known for many years, however in practice such kind of routing is applied unwillingly, and usually to equal cost paths. Because of the approximation properties of a typical link cost attributes, the delay seen by transport protocols vary when packets are forwarded uniformly on the paths and the paths are loaded differently. This phenomenon gives the effect similar to the one observed when route flapping occurs. Applying multipath routing when parallel path metrics are different (e.g. in EIGRP [4]) is very uncommon, and advantage of it is questionable. These problems can be alleviated by proper queuing or splitting traffic onto different paths using some hash function, however efficiency of such approach depends on the kind of traffic and requires significant processing and memory resources [13].

In the presented solution, we have merged delay approximation and multipath routing together with an appropriate packet forwarding algorithm, to achieve the desired goals. The proposed algorithm was inspired by the Nash equilibrium rule [20] and by the solution like MIRA (*Minimum Interference Routing Algorithm* [18]), but taking into account the mixed traffic (both TCP and UDP), unfortunately self-similar, heavy-tail one. The idea of this approach is to find the optimal paths for the transmissions in the whole network, sometimes at the cost of decreasing performance of some flows. Computation of the Nash equilibrium for the given network condition is very tiresome and hence unrealistic in real networks [20]. Moreover, it does not always lead to the optimal solutions [8]. The *off-line* solution like MIRA can be applied in the MPLS networks, where the declarations for the flows are known *a priori* as a result of the SLA contracts. However, such an approach is impractical in industrial (in fact - in any packet) networks, where both intensity and duration of flows are not known, and can vary by a few orders of magnitude.

## 4. Basic solution

As opposed to the typical method of the delay approximation [17], that relies on computing the average delay for every link in a constant interval, in HQRA algorithm we propose to apply an estimator – the first order low pass filter exactly the same, like the one used in TCP [14]:

$$a_{k+1} = \alpha_k x_k + (1 - \alpha_k)\, a_k \tag{1}$$

where $x_k$ is a link delay, i.e. sum of times of propagation, transmission, media access, processing and queuing. From among these elements, even if queues are pretty short, the most important is the queuing time – duration between enqueuing and dequeuing packets, independent on AQM algorithm. This is important not because of the duration itself, but because of its variation (see right subfigure 1). In equation (1), the $a_k$ is an estimate of the average link delay.

In such approach we encounter the problem of selecting the time constant of the filter (i.e. the coefficient $\alpha_k$). The optimal value of this constant depends on the kind of traffic, RTTs of particular flows, versions of TCP, statistical properties of traffic, etc. Therefore it is extremely difficult to give the general rule of setting up this value, if we want to achieve the fast reaction on the link load variation, together with limiting the number of the routing advertisements. So we have proposed the following adaptation method of the coefficient $\alpha_k$

$$\sigma_{k+1} = \beta\, |x_k - a_k| + (1 - \beta)\, \sigma_k \tag{2}$$

where $\sigma_k$ is an estimate of the average deviation (as analogue to the variable SDEV in the basic TCP algorithm [14]). The detailed discussion on the correct values of the coefficients $\alpha$ and $\beta$ can be found in [22–24].

The only link attribute used today in HQRA is the estimated link delay given by (1), however the $x_k$ can be interpreted as the value of a penalty function at the time $k$. Presented approach is inseparably linked with two general problems – the value $a_k$ changes very often, and advertising of the every change would generate the huge network load. A result of the delay estimation procedure described above is that obtaining exactly the same metrics on parallel paths is hardly possible. Therefore, the question arises — if, and if so, how, we should use particular paths. In other words – how to efficiently split the traffic among particular paths. The third serious problem is the stability of the set of equations (1), (2). It is easy to prove that the poles of transfer function of (1) and linearised (2) are in the unit disk for every $\alpha, \beta \in (0, 1)$, so the equations are locally stable. However, the prove of a stability of the equations (1), (2) in particular link does not guarantee stability of overall network. This is because the router knows exactly the state of its own link, knows approximated (but sometimes outdated) state of the links between current and traffic sink node, but this router knows nothing about the network that delivers packets to it, esp. the router knows nothing in what way, if any, the predecessor nodes react on the link attribute (and metrics) changes.

Considering extremely harmful flapping phenomenon, to achieve mentioned above, desired traffic properties, it was necessary to develop the algorithm of

- directing packets to alternative paths in ratio that depends on the route metrics ratio. This value must change gradually. This algorithm should work on the nodes that have more than one LFI path to the sink,

- link cost adaptation to the unknown conditions appearing in the part of the network that supply the current node,
- predicting the values of link cost attributes in such a way, that the changes of the link cost were advertised only if necessary.

## 5. Uncertainty of the link costs

All the algorithms mentioned above were developed on the basis of the formulation of the value of the link costs considered as the uncertain (fuzzy) values. We propose to define this uncertainty by splitting equation (2) into two other

$$\sigma^{+}_{k+1} = \beta\left(x_{k+1} - a_{k+1}\right) + (1-\beta)\,\sigma^{+}_{k} \quad \text{if } x_{k+1} > a_{k+1} \tag{3}$$

and

$$\sigma^{-}_{k+1} = \beta\left(a_{k+1} - x_{k+1}\right) + (1-\beta)\,\sigma^{-}_{k} \quad \text{if } a_{k+1} > x_{k+1} \tag{4}$$

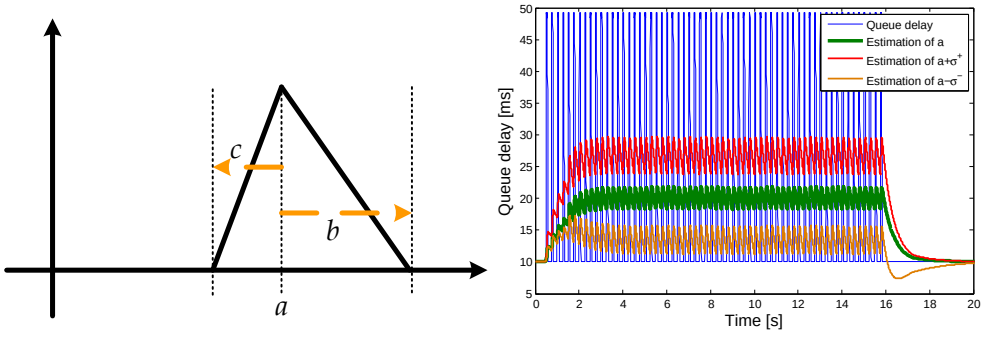



Fig. 1: Geometrical interpretation of uncertain value $\mathcal{A} = a^{b}_{c}$ (on the left) and results of the estimations (1), (3), (4) performed on a sample link.

The result of such approximation is presented in figure 1. Of course, the condition $\sigma_k = \sigma^{+}_{k} + \sigma^{-}_{k}$ is always satisfied, so equation (2) need not to be evaluated separately. If we compute the real values of deviations, we always have $\sigma^{+}_{k} = \sigma^{-}_{k}$, however it is not true in the case of estimates (3) and (4). These values can be used to obtain the uncertainty of the link delay. Because the link cost should be the integer value, we describe this cost as a

$$\mathcal{A}_k = [\rho a_k]^{\left[\rho\sigma^{+}_{k}\right]}_{\left[\rho\sigma^{-}_{k}\right]} \tag{5}$$

where $\rho$ is the coefficient that defines granulation and $\sigma^{+}_{k}$ is the uncertainty of increment, $\sigma^{-}_{k}$ is the uncertainty of decrement at the time $k$ (see geometrical interpretation on the left subfigure 1), and the operator $[\cdot]$ is rounding to the nearest integer. Operations on fuzzy numbers can be defined in different ways [34]. In the forthcoming considerations, for the clarity, the $\rho$ coefficient is not used. In our case, it is necessary to define the additive

operation (chaining attributes into metric) and the comparison operation (to split flows). The best way of the addition of uncertain numbers depends on the statistical properties of the traffic ($x_k$). These properties are time-varying, and therefore it is impossible to take them into account due to bandwidth limitations. A standard way of summing fuzzy numbers i.e. adding of central values and adding of uncertainties seems to be not suitable in such kind of application, because it leads to a fast increasing of the span of the uncertainty, making such metrics practically unusable. Therefore, we have chosen the following way of adding fuzzy numbers

$$\mathcal{A}+\mathcal{B}=a_y^x+b_t^s=\mathcal{C}=c_w^u=(a+b)^{\frac{ax+bs}{ab}}_{\frac{ay+bt}{ab}} \tag{6}$$

Such approach is equivalent to treating uncertainties as relative to the central value.

The second serious problem with uncertain values is their comparison. We have chosen the following approach

$$\begin{array}{rcl} a>b & \Rightarrow & \mathcal{A}>\mathcal{B} \\ a<b & \Rightarrow & \mathcal{A}<\mathcal{B} \\ a=b\wedge x+y>s+t & \Rightarrow & \mathcal{A}>\mathcal{B} \\ a=b\wedge x+y<s+t & \Rightarrow & \mathcal{A}<\mathcal{B} \end{array} \tag{7}$$

and we have defined the following coefficient for every LFI path, that describes overlapping ratio of two uncertain values

$$\lambda=\max\left(\frac{\min\left(a+x,b+t\right)+\max\left(a-y,b-t,0\right)}{x+y},0\right) \tag{8}$$

where particular values are defined like in equation (6), with the assumption $\mathcal{A}$ is the best (the least) metric, and $\mathcal{B}$ is the evaluated metric (if $\mathcal{A}=\mathcal{B}$ then $\lambda=1$). This coefficient is always in the range $\lambda\in[0,1]$, and coefficients $\lambda$ associated with particular paths are proportional to the probability of forwarding packet using the given path. Such an approach allows on unsophisticated splitting the traffic among paths, non quantized for faster links, and very simple to implement, even in hardware.

Moreover, if the path flow has low variation, this path will be chosen more willingly. Generally, the paths that have less but stable latency are preferred, and when traffic conditions get worse, the traffic is split among more paths, that satisfy the LFI condition.

## 6. Prediction and adaptation

In the previous papers we have presented several methods of traffic prediction. An incorrect prediction, together with the incorrect values of time coefficients in (1), (3), (4) leads to flapping phenomenon, and therefore is harmful.

The solution presented in [24] works satisfactory, but depends on several coefficients that have to be carefully tuned. This was criticised, and was considered as a weak point of

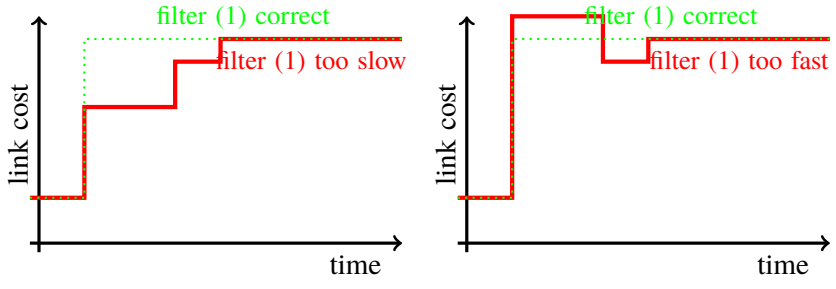



Fig. 2. Results of the incorrect chosen values of $\alpha$ (1). Only the main part of the metric is shown.

the solution. Moreover further investigation of this algorithm allows to observe the pathological inflation of the uncertainty in some (but very uncommon) cases, thus it incorrectly splits flows on multiple paths.

Therefore it was necessary to find simpler and less sensitive to coefficients way of traffic prediction. But the method of prediction is strictly related to proper adaptation of $\alpha$ and $\beta$. In the linearised version of the eq. set (1) and (2), the best properties (aperiodical, critical) are in the case when $\beta = \alpha$. This dependency was kept in simulations described in section 7. However empirical experiences of the TCP protocol require to think about different solutions. Results of keeping the incorrect values of the coefficients are presented on fig. 2.

Based on such observation we develop the algorithm that is presented on fig. 3 and formally described below. A signal to metric changes is created, when one of the following conditions occurs:

$$m_k + m_k^+ < a_k \wedge a_k^+ > a_k^- \tag{9a}$$

$$m_k - m_k^+ > a_k \wedge a_k^+ < a_k^- \tag{9b}$$

$$\frac{m_k^+ - a_k^+}{\max\left(m_k^+, a_k^+\right)} > \vartheta \tag{9c}$$

$$\frac{m_k^- - a_k^-}{\max\left(m_k^-, a_k^-\right)} > \vartheta \tag{9d}$$

where $\mathcal{M}_k = \left(m_{m^-}^{m^+}\right)_k$ is a link cost, and $\vartheta$ can be any value in a range $(0, 1)$, e.g. $\frac{1}{2}$.

As opposed to the solution presented previously, instead of changing and advertising the new link cost at the time $k$, when one of the conditions (9) are satisfied, the value of $a_k$ is stored and the analysis is delayed until time $k+\tau$, where $\tau$ can be any positive value. In our simulations $\tau = 1s$.

The new metric is **proposed** in the following way

$$m_{k+\tau} = \left[a_{k+\tau} + \phi\sigma_{k+\tau}\right], \tag{10a}$$

$$m^{+}_{k+\tau} = \left[a^{+}_{k+\tau}\right],\ m^{-}_{k+\tau} = \left[a^{-}_{k+\tau}\right], \tag{10b}$$

where $\phi = e^{-1}$ is chosen on the basis of the properties of the (1).

The algorithm attempts to change the link cost at the time $k+\tau$. However if at this time the conditions (9a i 9b) are not satisfied, then algorithm is terminated. This can be a result of the temporary congestion or fading. If these conditions are fulfilled permanently, the prediction error needs to be evaluated

$$E = m_k - a_{k+\tau} \tag{11}$$

Of course, if $E \neq 0$, then either some coefficients are chosen incorrectly or traffic statistics varies or both. Therefore the error is corrected by the equilibrium of the uncertainties:

$$\xi = E\frac{\sigma^{+}_{k+\tau} - \sigma^{-}_{k+\tau}}{\sigma_{k+\tau}} \tag{12}$$

If one of the condition is satisfied $\xi > 0 \wedge \sigma^{+}_{k+\tau} < \sigma^{-}_{k+\tau}$ or $\xi < 0 \wedge \sigma^{+}_{k+\tau} > \sigma^{-}_{k+\tau}$ this is a warning concerning the trend reversing. Therefore, the decision concerning the link cost change should be delayed by another $\tau$.

If trends are acknowledged, that the main part of the cost is assigned: $m_{k+\tau} = a_{k+\tau} + \xi$ and values of $\alpha$ (and $\beta$) should be corrected.

For any outgoing link, the previous cost change is recorded: $\delta_z = m_{z+1} - m_z$, where $z$ is a time of the change. This is used for evaluation how much the $\alpha$ (and $\beta$) differ from the optimal values (fig. 2). The new $\alpha$ i $\beta$ can be obtained using formulae:

$$\Xi = \begin{cases} 1 + \frac{\delta_k}{\delta_z}\exp\left(-\frac{k-z}{\Upsilon}\right) & \delta_k \neq 0 \wedge \delta_z \neq 0 \\ 1 & \delta_k = 0 \vee \delta_z = 0 \end{cases} \tag{13}$$

where $\Upsilon$ is some constant, and $k$ is the current time, and $z$ is time of the previous link cost change on the link, thus $k - z$ is the last interval between changes. If $(k - z) \to \infty$ then $\alpha$ and $\beta$ are close to optimal and $\Xi \to 1$. The $\Upsilon$ controls the speed of the convergence. In the simulations we have assumed $\Upsilon = 10s$. If the $\Upsilon$ is too small the formula (14) often is constrained. When (14) is constrained occasionally that is the sign of the correctness of $\Upsilon$. In the eq. (13), instead of $\exp(\cdot)$, any function f can be used, if f(0) $= 1$ and f monotonically decreases to zero. Actually, the new $\alpha$ can be obtained from:

$$\alpha_{k+1} = \begin{cases} C_l\alpha_k & \Xi < C_l \\ \Xi\alpha_k & C_l \leq \Xi \leq C_u \\ C_u\alpha_k & \Xi > C_u \end{cases} \tag{14}$$

where $C_l$ and $C_u$ are some constants. Theoretically it is necessary to make sure, $\alpha_k < 1$, but in practice it is not necessary, if the initial value of $\alpha$ is correct. In the simulations described in the section 7., $C_l = \frac{1}{2}$ and $C_u = 2$, however some results show that the

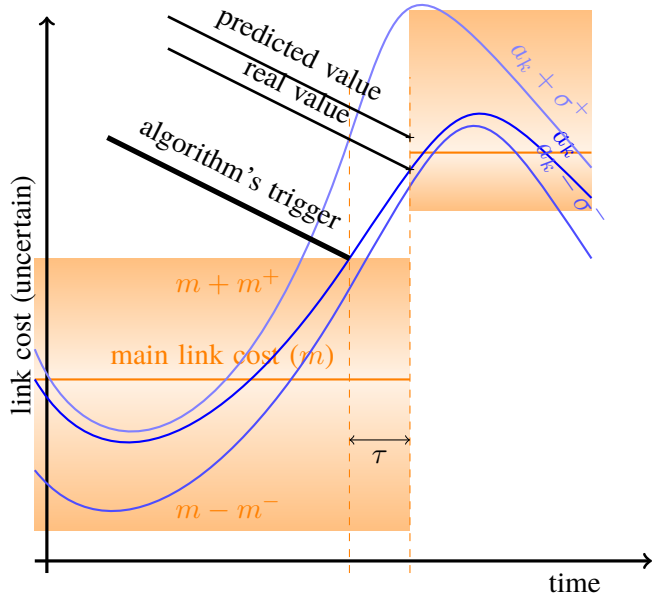



Fig. 3. Idea of the traffic prediction

constraint range can be wider. These constants influence on the speed of the algorithm's convergence, but too wide interval makes it more prone of the traffic pattern changes.

## 7. Simulations

The presented algorithm was tested using NS2 simulator [5] using network topology presented on figure 4. The impulse flow $F_1$ consists of the short packets send continuously with full speed of the outer links, then stops. This is the typical traffic for fieldbus networks. The average intensity of the flow is equal less or more half of the bandwidth of the inner links, and at the time 26s it doubles. The remaining flows are typical TCP flows with large packets. The very similar results can be obtained if these flows have self-similar statistics. The flow between nodes 2-3 ($F_2$) takes place between 5 and 29s, the flow between nodes 6-5 takes place between 17 and 40s of the simulation. Therefore the simulation has 5 phases – low intensity $F_1$ only, competition between $F_1$ and $F_2$, competition between $F_1$, $F_2$ and $F_3$, the same but intensity of $F_1$ doubles, competition between high intensity $F_1$ and $F_3$, and finally $F_1$ alone. The results of the simulations are presented on figure 5.

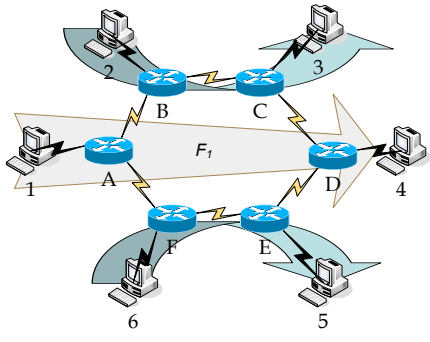



Fig. 4: Topology of the tested network. The flow $F_1$ is the impulse flow with very short messages typical for fieldbuses. The remaining flows are TCP flows. The outer (dark) links have ten times greater bandwidth and shorter delays than the inner (light) ones.

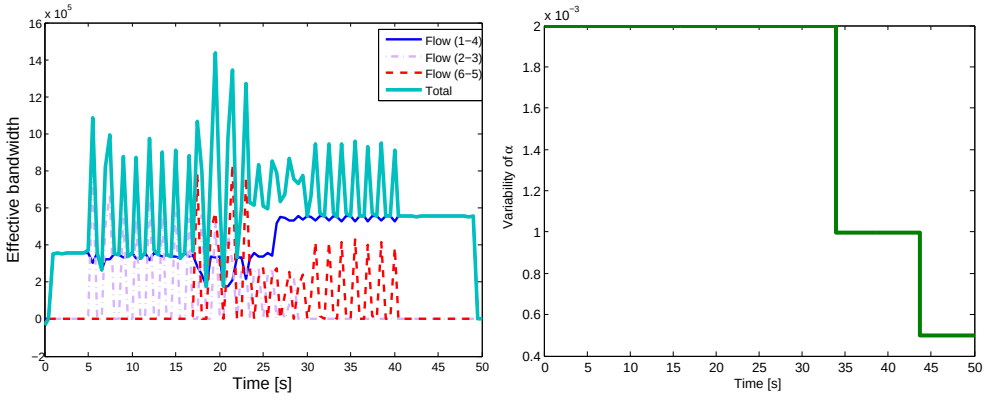



Fig. 5. On the left: average throughput for tested flows. On the right: Automatic selection of the best $\alpha$ (eq. 1).

## 8. Conclusions

The presented approach to the traffic engineering fits well in the multiple of requirements – including general TCP like, self-similar and real-time traffic, even mixed highly statistically differentiated traffic. The algorithm is low sensitive to the choosing of the initial values of the parameters and after some time it adapts well to any kind of a traffic. Applying of the algorithm causes $\frac{1}{5}$ to $\frac{1}{2}$ less packets dropped, depends on the intensity of traffic, and variability of the statistical pattern. Note, the inflating of the queue length and applying of the AQM can reduce the number of drops to the near zero, but by the cost of the increasing delays, therefore such solution was rejected. On the other hand it does not require high computational power, so can be easy applied in the low-cost nodes. The robustness of the algorithm was confirmed by the scenarios were the traffic patterns change faster then the adaptation ability, where adaptation was constrained by the coefficients $C_l$ and $C_u$. In particular, the algorithm fits well to the wireless links where link delays are highly variable in the noisy environment due to the link layer retransmissions, and can be easily adapted to other penalty functions than the delay of a link ($x_k$ in eq. 1).

## 9. Acknowledgements

This work has been supported by the Polish State Committee for Scientific Research (KBN) grant no. 3 T10A 037 28 (2005-2007).

## References

[1] [Online]. Available: http://www.ethernet-powerlink.org

[2] [Online]. Available: http://opcfoundation.org

[3] [Online]. Available: http://ieee1451.nist.gov

[4] "EIGRP Protocol." [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies˙tech˙note09186a0080093f07.shtml

[5] "Network simulator NS-2." [Online]. Available: http://www.isi.edu/nsnam/ns/

[6] ARINC, *ARINC Specification 629 Multi-transmitter data bus*, Aeronautical Radio INC, 2551 Riva Road Annapolis, Maryland, 1990.

[7] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Athena Scientific, Belmont, Massachusetts, 1998.

[8] A. Czumaj, "Selfish routing on the internet," Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA, Tech. Rep. grant CCR-0105701, May 15 2003. [Online]. Available: http://www.cs.njit.edu/˜czumaj/PUBLICATIONS/Selfish-Routing-Survey.pdf

[9] E.H.Callaway, *Wireless Sensor Networks: Architectures and Protocols*. CRC Press, 2004.

[10] R. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Transactions On Communication*, vol. 25, no. 1, pp. 73–84, January 1977.

[11] J. J. Garcia-Luna-Aceves, S. Vutukury, and W. T. Zaumen, "A practical approach to minimizing delays in internet routing protocols," in *IEEE ICC*, Vancouver, Canada, June 6–10 1999.

[12] A. Grzech, *Sterowanie ruchem w sieciach teleinformatycznych*. Oficyna wydawnicza Politechniki Wrocławskiej, 2002, in polish.

[13] C. Hopps, "Analysis of an equal-cost multi-path algorithm," RFC, Tech. Rep. 2992, November 2000.

[14] V. Jacobson, "Modified TCP congestion avoidance algorithm," end2end-interest mailing list, Tech. Rep., 30 April 1990.

[15] L.-E. Jonsson, G. Pelletier, and K. Sandlund, "RObust Header Compression (ROHC): A link-layer assisted profile for IP/UDP/RTP," RFC, Tech. Rep. 4362, January 2006. [Online]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc4362.txt

[16] M. Juszczak, M. Morawski, and A. Bartoszewicz, "Multipath routing in mpls networks – a survey," *Journal of Applied Computer Sciences (JACS)*, no. 1, pp. 7–30, 2006.

[17] A. Khanna and J. Zinky, "The revised ARPANET routing metric," in *SIGCOMM*. ACM, August 1989, pp. 45–56.

[18] M. Kodialam and T. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *INFOCOM*, Tel Aviv, Israel, MArch 2000, p. 884–893.

[19] H. Kopetz, *Real-Time Systems. Design Principles for Distributed Embedded Applications*, 2nd ed., ser. The Kluwer international Series in engineering and computer science, Real-Time Systems. Boston/Dordrecht/London: Kluwer Academic Publishers, 1998.

[20] R. La and V. Anatharam, "Optimal routing control: Game theoretic approach," in *36th conference on decision and Control*, San Diego, USA, December 1997, pp. 2910–2915.

[21] R. Mitchell, *Profibus: a pocket guide*. ISA, 2004.

[22] M. Morawski, "Multipath routing that supports QoS," in *Sieci Komputerowe*, Zakopane, Poland, 24–26 June 2005, pp. 171–182.

[23] ——, "Uncertain metrics applied to QoS multipath routing," in *5th International Workshop on Design of Reliable Communication Networks, DRCN'05*, Island of Ischia, Naples - Italy, 16-19 October 2005, pp. 353–360.

[24] ——, "Optimal adaptive routing with efficient flapping prevention," in *Proceedings of 4th Polish-German Teletraffic Symposium PGTS*, Wrocław, Poland, 20–21 September 2006, pp. 85–94.

[25] C. Murthy and B. Manoj, *Ad-Hoc Wireless Networks*. Prentice Hall Communication, 2004.

[26] K. Åström and B. Wittenmark, *Computer-controlled systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[27] H. G. Perros, *Connection-oriented Networks SONET/SDH, ATM, MPLS and OPTICAL NETWORKS*. John Wiley & Sons Inc,, 2005.

[28] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhäuser, 2003.

[29] H. Thompson, "Wireless and internet communication technologies for monitoring and control," *Control Engineering Practice*, vol. 12, pp. 781–791, 2004.

[30] S. Vutukury and J. J. Garcia-Luna-Aceves, "A practical framework for minimum-delay routing in computer networks," *Journal of High Speed Networks*, vol. 8, no. 4, pp. 241–263, 1999.

[31] ——, "A simple approximation to minimum-delay routing," in *ACM SIGCOMM*, Cambridge, Massachusetts, September 1—3 1999.

[32] ——, "A traffic engineering approach based on minimum-delay routing," in *IEEE IC3N*, Las Vegas, Nevada, USA, October 16–19 2000.

[33] ——, "MPATH: a loop-free multipath routing algorithm," *Journal of Microprocessors and Microsystems (Elsevier)*, vol. 24, pp. 319–327, 2001.

[34] O. Wolkenhauer, *Data Engineering: Fuzzy Mathematics in Systems Theory and Data Analysis.* John Wiley & Sons, Inc, 2001.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 165–175

# Mathematical models for combined OSPF/MPLS routing optimization in IP networks*

MICHAŁ ZAGOŻDŻON [a,b]

[a]Institute of Telecommunications
Warsaw University of Technology
mzagozdz@tele.pw.edu.pl

[b]Research and Development Center
Telecom Poland

**Abstract:** In this paper we address intra-domain routing optimization issues in Internet networks related to a combination of Open Shortest Path First (OSPF) routing protocol and Multi Protocol Label Switching (MPLS) technology. Since MPLS can be combined with any routing protocol there has appeared an idea of coupling IP streams routed via the OSPF protocol and MPLS Label Switched Paths (LSPs) in the same network. In this approach some of the flows are routed by means of OSPF while the others are carried on predefined MPLS label switched paths (LSP). Such combined routing is potentially more flexible and efficient than OSPF itself from the traffic engineering (TE) viewpoint. In the paper we present mathematical models for three scenarios of combining OSPF with MPLS technique in a single IP network, which are: *Overlay* model, *Basic IGP Shortcut* and *MPLS Based TE Links* models and draw our future research directions in this area.

**Keywords:** IP, combined IGP/MPLS routing, OSPF, MPLS

## 1. Introduction

The OSPF protocol is widely used in IP networks as an IGP intra-domain routing protocol. The protocol belongs to the group of the link state routing protocols. This in particular means that some information describing link state, such as link status (e.g., working or failed), link utilization and link delay are taken into account when traffic routes are selected. The OSPF routing is based on the shortest paths computations. Each link is assigned a positive integer metric (called administrative weight). Such a metric is set by the network

*

The paper was written under Polish research grant N N517 4396 33 "Projektowanie hybrydowych sieci IP" sponsored by Polish State Committee for Scientific Research.

administrator and can either be fixed or can reflect the current link state [11]. In our work we assumed that link metrics are fixed. A routing optimization problem related to OSPF can be roughly stated as follows: given traffic demands and a network topology, find a system of link metrics such that the resulting shortest path routing is feasible within the available link capacities (i.e., that the resulting link loads do not exceed link capacities). In the case when there are multiple shortest paths between a pair of source-destination nodes, the routers apply the ECMP (Equal Cost Multiple Path) rule that equally splits the traffic flow leaving the router among all the outgoing links belonging to the shortest paths to the given destination.

The OSPF/ECMP routing optimization problem is *NP*-complete [18]. Because it is an allocation problem, one can extend it with an auxiliary objective function that will assure that the links are utilized in a reasonable way, on top of the basic requirement that the links are not overloaded. For example, one may use an objective function that forces the metrics to maximize the residual (unused) capacity, since the unused capacity can be utilized for flow restoration in the case of network failures.

Flow allocation problems related to OSPF/ECMP have been addressed by several researchers who suggested different resolution methods. In [18, 24, 3, 7, 2] the authors focus on the exact approach based on mixed-integer programming (MIP) problem formulations solved by the branch-and-cut (B&C) approach. They propose several classes of so called *valid inequalities* used to strengthen the problem formulation in order to improve the B&C algorithm efficiency. The considered problem is intrinsically very difficult and therefore it frequently happens that even for medium size networks it is virtually impossible to solve the MIP formulation of the problem to optimality. Therefore heuristic methods are of interest—such heuristics can be based on evolutionary algorithms (EA) [12, 9, 20], simulated annealing (SAN) [21], simulated allocation [6], local-search heuristics [4, 9, 5], and different combinations of these [19]. Heuristic methods can provide reasonable approximate solutions in a reasonable time, and can be used as upper bounds for the B&C solvers. A disadvantage of using heuristics alone is that in general it is not known how far are their solutions from the unknown global optimum (sometimes they can be very far from optimum).

Besides the computational problems associated with OSPF routing optimization, there is another potential drawback associated with this type of routing. Although the destination-based, shortest-path routing principle (as used in OSPF) is relatively simple to implement and assures reasonable traffic performance, its flexibility in using routes within a domain is limited: the use of link metrics combined with the ECMP can substantially constrain the set of traffic flows that can be actually realized in the network. Hence, a situation can arise when feasible traffic flow patterns theoretically exist, still such patterns cannot be achieved with an OSPF type of routing.

We are convinced that many of the issues associated with the OSPF routing problem complexity and its non-flexibility in terms of realizable flow patterns can be overcome by combining the OSPF routing with the Multi-Protocol Label Switching (MPLS) technology. By doing this we can preserve simplicity of OSPF (a destination-based shortest-path routing) and gain from flexibility of MPLS (origin-destination explicit path routing). MPLS provides

a mechanism for fast packet switching in the IP routers by coloring packets belonging to the same IP FECs (Forwarding Equivalence Classes), and each router chooses an outgoing link interface on the basis of the packet color. The MPLS protocol requires that the network administrator determines traffic routes in advance, or new LSPs are created using information obtained from the routing protocol.

In the context of the combined OPSF/MPLS routing we can define several rules determining their coexistence and cooperation, making several variants of combined routing possible. In the literature four specific models can be found, namely: (1) *Basic IGP Shortcut* and (2) *IGP Shortcut* proposed by [15], (3) *MPLS Based TE (Traffic Engineered) Links* proposed by [14] and an (4) *Overlay* model. Many authors have tackled problems of routing optimization in networks for these types of combined OSPF/MPLS routing [13, 21, 10, 17], but in their work they focus only on some of the variants and their conclusions are based on the results obtained by solving problems using different heuristic methods which base on problem decomposition. As, in general, heuristic methods may deliver solutions far from optimum, in our research activities we focused on the exact methods, such as *branch-and-cut*. Another advantage of using exact methods is, that although they may not provide optimal solutions, they can be used to compute close to optimum solutions of known quality. To solve the problem using a branch-and-cut solver (e.g. CPLEX [8]) we must formulate them as Mixed Integer Programming problems (MIPs). Thus, in the paper we first informally present three following hybrid routing variants: *Overlay* model, *Basic IGP Shortcut* and *MPLS Based TE Links*, and then in Sections (3.1-3) we give their strict mathematical formulations. The paper is summarized with some conclusions and plans for the future research.

## 2. Hybrid routing variants

First hybrid routing variant under consideration is an ***Overlay*** model. It assumes two distinct groups of traffic demand requirements. The demands belonging to one group are forwarded using the shortest paths computed by OSPF, while the demands of the other group use the predefined LSPs. This model assumes creating an MPLS overlay in which a selected subset of demands uses the *tunnels* (i.e., Explicitly Routed LSPs) created along the IP links and not obeying the destination-based shortest-path rule. It follows that a hybrid routing optimization problem consists in determining how the demands should be spilt between the two groups, and solving a routing optimization problem for each demand group: a metric setting problem for the OSPF demands, and a single-path routing problem for the MPLS demands.

An MPLS demand can in general be realized in parallel on more than one tunnel. We will not exploit this option (although it simplifies the mathematical treatment) because the implementation of bifurcated flows (frequently called load balancing in this context) may vary in technological solutions of different vendors and has not been standardized in MPLS RFCs as yet ([22]). The quality of service (QoS) requirements may also determine the distribution of the demands between the OSPF and MPLS groups, for example we may require

that some demands are VPN connections to be realized as LSPs; also, some demands may require short failure recovery times possible to achieve using for example the Fast Rerouting mechanism of MPLS. Moreover, when using MPLS LSPs we can measure the volumes of the end-to-end traffic. Estimation of the demand matrix of the flows realized using destination-based routing, like OSPF, is much less accurate in this aspect because it is mainly based on the information about the link loads in the network and their cross-correlation [16, 25].

Another variant, being a generalized version of an *Overlay* model is a ***Basic IGP Shortcut***. In this scenario, each router in the IP network before sending any packet to a given destination checks whether there is a tunnel originating in the router and destined to the packets destination. If yes, the router adds a label associated with the tunnel in front of the packet IP header and forwards it using this tunnel. If there is no such tunnel, packet follows classical IGP routing. Basic IGP Shortcut is a generalization of an Overlay model, because for a given tunnel, not only demands originating in tunnels ingress node (say router A) and destined to the egress node (say router B) are forwarded using this tunnel, but also all other pockets which on their way to B reached router A.

Last scenario considered in this paper is referred to as an ***MPLS Based TE Links***. In this model MPLS tunnels are advertised throughout the IGP domain as IP TE (Traffic Engineered) links. After setting up an MPLS tunnel, its ingress router creates on its basis a virtual interface (link) and after assigning a weight, floods the information about newly created link in the IGP domain. The route of the tunnel and the assigned weight can be set either dynamically or by the network administrator. Other routers, as in the case of physical IP links, after receiving the information about newly created IP TE links use their weights in the shortest path calculations. As we can see, by introducing new virtual IP TE links we can modify the shortest path routes to some destinations. Of course, old routes can be only substituted with shorter ones.

The presented combined routing variants implie that certain routers need to be equipped with both MPLS and IP switching capabilities. Each packet traversing the network receives a protocol tag and according to that is served either in the MPLS mode or in the IP mode. Of course, in the network there can be routers with only MPLS or IP switching capability. In the former case, the router can only switch MPLS tags, while in the latter, the router can only switch packets on the basis of their destination address. Still, to some extent, all the described variants can be implemented and used in such a networks.

## 3. Mathematical problem formulations

The considered network models consists of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$ with the set of nodes $\mathcal{V}$ representing routers, the set of (directed) links $\mathcal{E}$ ($\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$), and the set of (directed) demands $\mathcal{D}$ ($\mathcal{D} \subseteq \mathcal{V} \times \mathcal{V}$). The number of nodes, links, and demands is equal to $V = |\mathcal{V}|$, $E = |\mathcal{E}|$, $D = |\mathcal{D}|$, respectively. For each link $e \in \mathcal{E}$ its capacity is denoted by $c_e$, and $a(e)$ and $b(e)$ denote its originating and terminating node. For a a given node $v \in \mathcal{V}$ we define the set of links $\delta^+(v)$ outgoing from node $v$, and the set of links $\delta^-(v)$ incoming to

node $v$. More precisely, $\delta^+(v) = \{e \in \mathcal{E} : a(e) = v\}$ and $\delta^-(v) = \{e \in \mathcal{E} : b(e) = v\}$.

The set $\mathcal{D}$ represents traffic demands between pairs of nodes. Without loss of generality, we assume that there are the demand requirements between all pairs of nodes ($\mathcal{D} = \{(v,t) : v \in \mathcal{V}, t \in \mathcal{V}, v \neq t\}$. $h_{vt}$ is the traffic volume of demand originating in node $v$ and destined to node $t$, expressed in the same units of bandwidth as capacity of links. Constant $H^t$ will denote the summary flow destined to node $t$ ($H^t = \sum_{v \in \mathcal{V} \setminus \{t\}} h_{vt}$). In the sequel we will always assume that link capacities $\boldsymbol{c} = (c_e : e \in \mathcal{E})$ and demand volumes $\boldsymbol{h} = (h_{vt} : (v,t) \in (v \in \mathcal{V}, t \in \mathcal{V} \setminus \{v\})$ are fixed and given.

As far as the modeling of MPLS tunnels is concerned, we can sometimes limit ourselves to the predefined set of all admissible paths $\boldsymbol{\mathcal{P}}$ that can be used to carry traffic. Each (simple) path $p \in \boldsymbol{\mathcal{P}}$ is represented by its set of links denoted by $\mathcal{E}_p$ ($\mathcal{E}_p \subseteq \mathcal{E}$). For each pair of nodes $(v,t) :\ v \in \mathcal{V}, t \in \mathcal{V}, v \neq t$ the set of the admissible paths that can be set-up from $v$ to $t$ is denoted by $\mathcal{P}_{vt}$ ($\mathcal{P}_{vt} \subseteq \boldsymbol{\mathcal{P}}$). Similarly, the set of all admissible paths in $\boldsymbol{\mathcal{P}}$ that use a particular link $e \in \mathcal{E}$ will be denoted by $\mathcal{P}_e$, i.e., $\mathcal{P}_e = \{p \in \boldsymbol{\mathcal{P}} : \mathcal{E}_p \ni e\}$. Finally, for each pair of nodes $(v,t) :\ v \in \mathcal{V}, t \in \mathcal{V}, v \neq t$ and each link $e \in \mathcal{E}$ we introduce the set of all admissible paths originating in node $v$ and terminating in $t$ that traverse link $e$: $\mathcal{P}_{evt} = \mathcal{P}_e \cap \mathcal{P}_{vt}$. Clearly, $\boldsymbol{\mathcal{P}} = \bigcup_{v \in \mathcal{V}, t \in \mathcal{V}, v \neq t} \mathcal{P}_{vt} = \bigcup_{e \in E} \mathcal{P}_e = \bigcup_{e \in \mathcal{E}} \bigcup_{v \in \mathcal{V}, t \in \mathcal{V}, v \neq t} \mathcal{P}_{evt}$.

### 3.1. Overlay model

Below we present a problem of routing design in IP network with an *Overlay* OSPF/MPLS type of routing formulated as a Mixed Integer Programme. We first present and explain variables, then give the objective and constraints used:

**variables**

| | |
|---|---|
| $z$ | minimal value of residual capacity common to all links |
| $w_e$ | metric (weights) assigned to link $e$ (integer) |
| $r_{vt}$ | length of the shortest-path from $v$ to $t$, ($r_{vv} = 0$) |
| $x_{et}$ | volume of the aggregated flow destined to node $t$ on link $e$ |
| $y_{vt}$ | common value of non-zero flow leaving node $v$ for all the flows destined to $t$ assigned to links outgoing from $v$ and belonging to the shortest-paths to $t$ |
| $u_{et}$ | binary variable equal to 1 iff link $e$ is on a shortest-path to node $t$; 0 otherwise |
| $o_{vt}$ | binary variable equal to 1 iff demand from $v$ to $t$ is realized using shortest path routing; 0 if using an MPLS tunnel, ($o_{vv} = 0$) |
| $u_{vtp}$ | binary variable equal to 1 iff demand from $v$ to $t$ is realized on MPLS path $p$; 0 otherwise |

**objective**

maximize $z$

**constraints**

$$\sum_{p\in\mathcal{P}_{vt}} u_{vtp} = 1 - o_{vt} \qquad v,t \in \mathcal{V}, \quad v \neq t \tag{1a}$$

$$\sum_{e\in\delta^+(v)} x_{et} - \sum_{e\in\delta^-(v)} x_{et} = o_{vt} h_{vt} \qquad v,t \in \mathcal{V}, \quad v \neq t \tag{1b}$$

$$\sum_{t\in\mathcal{V}} (x_{et} + \sum_{v\in\mathcal{V}\setminus\{t\}} \sum_{p\in\mathcal{P}_{evt}} u_{vtp} h_{vt}) + z \leq c_e \; e \in \mathcal{E} \tag{1c}$$

$$0 \leq y_{a(e)t} - x_{et} \leq (1 - u_{et}) H^t \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \tag{1d}$$

$$x_{et} \leq u_{et} H^t \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \tag{1e}$$

$$1 - u_{et} \leq r_{b(e)t} + w_e - r_{a(e)t} \leq (1 - u_{et}) M \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \tag{1f}$$

$$w_e \geq 1 \qquad e \in \mathcal{E} \tag{1g}$$

$$\sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{V}\setminus\{v\}} (1 - o_{vt}) \leq \beta |D|. \tag{1h}$$

Constraint (1a) assures that demand originating in node $v$ and destined to node $t$ is either realized using OSPF shortest path forwarding ($o_{vt} = 1$) or uses one of the predefined LSPs ($o_{vt} = 0$). As $u_{vtp}$ variables are binary, we also require that demands must be realized on a single LSPs. Constraint (1h) sets an upper bound on the number of demands realized using MPLS shortcuts, where $0 \leq \beta \leq 1$ is a parameter. Note that in fact this value should be kept as small as possible so parameter $\beta$ could be made a variable to be minimized.

To model flows in the network we use variables $x_{et}$ specifying the amount of aggregated flow on link $e \in \mathcal{E}$ for the traffic destined to (a remote) node $t \in \mathcal{V}$. Using these variables and applying in each transit node $v$ the flow conservation law for the aggregated flows we write down Constraint (1b).

In a network with shortest-path OSPF-like routing, flows are forwarded on the shortest paths from source to sink nodes. The paths are calculated in each router with respect to integral link metrics which are variables in our optimization problem: $\boldsymbol{w} = (w_1, w_2, \ldots, w_{|\mathcal{E}|})$, where $w_e \in \{1, 2, \ldots, 65535\}, e \in \mathcal{E}$. To make the $x_{et}$ flows consistent with shortest path OSPF/ECMP routing flows need to be split in nodes according to ECMP rule (Constraint (1d)) and may only follow shortest paths (Constraint (1e)) which are unique (Constraint (1f)). For this purpose we introduce binary variables $u_{et}$ equal to 1 if and only if link $e$ belongs to the shortest path to node $t$, 0 otherwise (see [18] for details).

Constraint (1d) assures that if link $e$ belongs to one of the shortest paths from node $a(e)$ to node $t$ then its carried flow destined to node $t$ is equal to $y_{a(e)t}$ – a continuous value common to all links outgoing from node $a(e)$ and belonging to the shortest paths to destination $t$. The continuous variables $r_{vt}$ are used to express the length of the shortest path from node $v$ to node $t$. $M$ is a big integer value. Finally, in Constraint (1c) we assure that the overall link loads must not exceed the link capacities. Variable $z$ is unrestricted in sign and is associated with the value of the residual capacity available on the links.

Our objective function maximizing $z$ will provide the maximum residual capacity possible to achieve simultaneously on all links. Note that the optimization problem (1) does always have a feasible solution, and in the case when it is not possible to realize the demands without exceeding capacities of some links, the maximal value of $z$ will be negative and then $-z$ will denote the minimum value of additional capacity required to be installed on at least

one link to make the allocation feasible. Certainly, other objective functions could be used in Problem (1) (for example the piece-wise linear penalty function of [4]). This, however, would not influence our considerations.

### 3.2. Basic IGP Shortcut

In this section we will formulate the next hybrid routing variant under consideration, which is *Basic IGP Shortcut*. The list of variables does not contain variables which were presented in the context of Problem (1). The formulation is presented below:

**new variables**

$x_{vt}$ aggregated flow leaving node $v$ and realized on MPLS shortcut to node $t$

$x_{vtp}$ continuous flow on path/shortcut $p$ from node $v$ to node $t$

$u_{vtp}$ binary variable equal to 1 iff path $p$ is being used as a shortcut from node $v$ to node $t$; 0 otherwise

**objective**

maximize $z$

**constraints**

$$x_{vt} + \sum_{e\in\delta^+(v)} x_{et} - \sum_{e\in\delta^-(v)} x_{et} = h_{vt} \qquad v,t \in \mathcal{V}, v \neq t \qquad \text{(2a)}$$

$$\sum_{t\in\mathcal{V}} (x_{et} + \sum_{v\in\mathcal{V}\setminus\{t\}} \sum_{p\in\mathcal{P}_{evt}} x_{vtp}) + z \leq c_e \qquad e \in \mathcal{E} \qquad \text{(2b)}$$

$$0 \leq y_{a(e)t} - x_{et} \leq (1 - u_{et})H^t \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \qquad \text{(2c)}$$

$$x_{et} \leq u_{et} H^t \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \qquad \text{(2d)}$$

$$1 - u_{et} \leq r_{b(e)t} + w_e - r_{a(e)t} \leq (1 - u_{et})M \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \qquad \text{(2e)}$$

$$w_e \geq 1 \qquad e \in \mathcal{E} \qquad \text{(2f)}$$

$$\sum_{e\in\delta^+(v)} x_{et} \leq (1 - \sum_{p\in\mathcal{P}_{vt}} u_{vtp})H^t \qquad v,t \in \mathcal{V}, v \neq t \qquad \text{(2g)}$$

$$\sum_{p\in\mathcal{P}_{vt}} x_{vtp} = x_{vt} \qquad v,t \in \mathcal{V}, v \neq t \qquad \text{(2h)}$$

$$x_{vtp} \leq u_{vtp} H^t \qquad v,t \in \mathcal{V}, v \neq t, p \in \mathcal{P}_{vt} \qquad \text{(2i)}$$

$$\sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{V}\setminus\{v\}} \sum_{p\in\mathcal{P}_{vt}} u_{vtp} \leq \beta|\mathcal{V}|^2 \qquad \text{(2j)}$$

New variable $x_{vt}$ is used to express the summary flow leaving node $v$ and destined to node $t$ that is chosen to be realized using an MPLS shortcut. For this purpose we include this variable in the flow conservation Constraint (2a) treating it as if it was a link originating in node $v$ and terminating in note $t$. Summary flow $x_{vt}$ can be realized using exactly one of the candidate paths $p \in \mathcal{P}_{vt}$. This is assured by Constraints (2h) , (2i), and (2g). Constraint (2g) also says, that summary flow routed in OSPF domain leaving certain node $v$ and destined to $t$ ( $\sum_{e\in\delta^+(v)} x_{et}$) must be zero if at least one $u_{vtp}$ for $p \in \mathcal{P}_{vt}$ is equal to 1. On the other hand if this summary flow is greater than zero, than all the $u_{vtp} = 0$ for $p \in \mathcal{P}_{vt}$ - what means that we cannot use any shortcut from $v$ to $t$ and $x_{vt} = 0$. Last Constraint (2j), similar to Constraint (1h) sets an upper bound on the number of the shortcuts that can be set-up in the network.

### 3.3. MPLS Based TE Links

In this section we are presenting the MIP formulation of the last considered hybrid routing variant called *MPLS Based TE Links*. Problem formulation uses a slightly modified notation which was presented in Section (3.) Here the set of all IP links $\mathcal{E}$ is divided into two subsets: the set of physical IP links $\mathcal{E}'$ and the set of candidate IP TE links ($\mathcal{E}''$) that may be dynamically set-up as a virtual interfaces/links. This sets are disjoined and $\mathcal{E} = \mathcal{E}' \cup \mathcal{E}''$. Weights are assigned to the links of both types. By $\mathcal{P}_{e''}$ we denote the set of candidate LSPs that can be used for the realization of TE links $e'' \in \mathcal{E}''$. Clearly, all paths from $p \in \mathcal{P}_{e''}$ should have their ingress and egress nodes in $a(e'')$ and $b(e'')$ respectively. Finally, for each TE link $e'' \in \mathcal{E}''$ and each physical IP link $e' \in \mathcal{E}'$ we introduce the set of all admissible paths $p \in \mathcal{P}_{e''}$ that traverse link $e'$: $P_{e''e'}$.

**modified indices**

| | |
|---|---|
| $e' \in \mathcal{E}'$ | physical IP links |
| $e'' \in \mathcal{E}''$ | candidate IP TE links |
| $e \in \mathcal{E}$ | IP links ($\mathcal{E} = \mathcal{E}' \cup \mathcal{E}''; \quad \mathcal{E}' \cap \mathcal{E}'' = \emptyset$) |
| $p \in \mathcal{P}_{e''}$ | candidate MPLS paths/tunnels used for TE links realization |

**new variables**

| | |
|---|---|
| $x_{e''p}$ | continuous flow on MPLS path $p$ realizing TE link $e''$ |
| $u_{e''p}$ | binary variable equal to 1 iff path $p$ is used to realize TE link $e''$; 0 otherwise |
| $y_{e''}$ | capacity of TE link $e''$ |

**objective**

maximize $z$

**constraints**

$$\sum_{e\in\delta^+(v)} x_{et} - \sum_{e\in\delta^-(v)} x_{et} = h_{vt} \qquad t \in \mathcal{V}, v \in \mathcal{V}, v \neq t \qquad (3a)$$

$$\sum_t x_{e't} + \sum_{e''\in\mathcal{E}''} \sum_{p\in\mathcal{P}_{e''e'}} x_{e''p} + z \leq c_{e'} \qquad e' \in \mathcal{E}' \qquad (3b)$$

$$\sum_t x_{e''t} \leq y_{e''} \qquad e'' \in \mathcal{E}'' \qquad (3c)$$

$$0 \leq y_{a(e)t} - x_{et} \leq (1 - u_{et}) \sum_v h_{vt} \qquad t \in \mathcal{V}, e \in \mathcal{E} \qquad (3d)$$

$$x_{et} \leq u_{et} \sum_v h_{vt} \qquad t \in \mathcal{V}, \quad e \in \mathcal{E} \qquad (3e)$$

$$1 - u_{et} \leq r_{b(e)t} + w_e - r_{a(e)t} \leq (1 - u_{et})M \qquad t \in \mathcal{V}, e \in \mathcal{E} \qquad (3f)$$

$$w_e \geq 1 \qquad e \in \mathcal{E} \qquad (3g)$$

$$\sum_p x_{e''p} = y_{e''} \qquad e'' \in \mathcal{E}'' \qquad (3h)$$

$$x_{e''p} \leq u_{e''p} M \qquad e'' \in \mathcal{E}'', p \in \mathcal{P}_{e''} \qquad (3i)$$

$$\sum_p u_{e''p} \leq 1 \qquad e'' \in \mathcal{E}'' \qquad (3j)$$

$$\sum_{e''\in\mathcal{E}''} \sum_p u_{e''p} \leq \beta |\mathcal{E}''| \qquad (3k)$$

In this case, the problem formulation contains two capacity constraints, related to physical IP links (Constraint (3b)) and MPLS based TE links (Constraint (3c)). The capacities of

the latter denoted by $y_{e''}$ are provided by means of the bandwidth assigned to MPLS LSPs realizing each installed TE link $e''$. In our model we assume, that MPLS LSPs can only use resources of physical IP links. Again, we require that at most one LSP can be used to create every link $e'' \in \mathcal{E}''$ (Constraints (3h), (3i) and (3j)). Upper bound on the on the number of IP TE links that can be installed in the network is imposed by Constraint (3k).

### 3.4. Exact resolution methods

Hybrid routing optimization problems (1), (2) and (3) presented in the preceding sections can be solved using any MIP solver. The problems can be further simplified by dropping the integrality requirement for metric variables $\boldsymbol{w}$ since we can in most of the cases multiply the optimized fractional weights by their smallest common denominator. Bounding variables $\boldsymbol{w}$ through the constraints $w_e \leq W, \quad e \in \mathcal{E}$ can also help. As the reader may have already noticed, instead of introducing the list of candidate paths $P_{vt}$ (Problems (1) and (2)) or $P_{e''}$ (Problem (3)) we could use a node-link notation, which from its nature takes into account all the possible paths. This would, however, significantly increase the complexity of our models. The limitation of the predefined sets of candidate paths can be overcome by introducing the *Path Generation* technique to the branch-and-bound algorithm (see [18] and [1]). We will study the effectiveness of this approach in our future research. In parallel we will investigate on the application of the branch-and-cut approach where the problem formulation can be strengthened by adding so called *valid inequalities*. This in fact would be the continuation of the work which results were presented in [23]

## 4. Final Remarks

In this paper we have studied an optimization problems related to hybrid OSPF/MPLS routing in IP networks. The considered routing models assume that the majority of traffic flows is realized by the standard OSPF switching capability with the ECMP splitting rule, while the remainder of the flows are sent through MPLS tunnels. We have presented three scenarios of OSPF and MPLS coexistence and formulated associated routing optimization problems as Mixed Integer Programmes. They will form the basis for the exact methods that will be elaborated in the future. This exact methods will be mainly based on the branch-and-bound-and-cut algorithm and will be used for the evaluation of the effectiveness of the presented combined routing variants in terms of meeting the Traffic Engineering objectives. As an exact methods may not be applicable for large problem instances, in the next step, a heuristic algorithms taking advantage of the problem decomposition will be considered. An example of such algorithm might be a two phase approach, in which in Phase I we compute a reasonable OSPF weight system using one of the following meta-heuristics: Local Search, Simulated Annealing or Evolutionary Algorithm and in Phase II we try to *tune* the network by adding an arbitrary number of tunnels. Our initial numerical experiments have shown, that the value of the objective function for the solution of Phase I can be significantly improved when using a small number of MPLS LSPs in Phase II.

Another interesting approach, being also a two-phase method assumes that in Phase I we are solving a simple single path allocation problem in order to find a traffic flow pattern that optimizes our objective function and in Phase II we are trying to find the link weights such that most of the paths are also the single shortest paths. If certain path is at the same time the shortest path between some two end nodes, then there is no need to set-up an LSP, because the traffic will also follow this path if routed in OSPF domain. This is the simple example explaining the idea of how such two-phase method could work. Its improved versions will be also considered and used as a comparison basis in our future research.

Our preliminary numerical experiments show that the hybrid approach allows not only for a better resource utilization, but also makes the associated hybrid routing optimization problem easier to solve. Even when using simple weight systems like RIP or Inverse Capacity Proportional Weights which generally do not ensure good quality solutions we can introduce significant improvements by combining OSPF with MPLS. Therefore, a hybrid OSPF/MPLS routing schemes presented in the paper seems to be an attractive alternative as compared to the solutions based only on shortest path routing schemes as, in particular, OSPF.

## References


[1] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[2] A. Bley, M. Grötschel, and R. Wessäly. Design of broadband virtual private networks: Model and heuristics for the b-win. In *Proc. DIMACS Workshop on Robust Communication Networks and Survivability, AMS-DIMACS Series, 53*, pages 1–16, 1998.

[3] N. Bourquia, W. Ben Ameur, E. Gourdin, and P. Tolla. Optimal shortest path routing for Internet networks. In *Proceedings INOC'2003, Evry-Paris*, pages 119–125, 2003.

[4] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. *Proceedings of IEEE INFOCOM (2)*, pages 519–528, 2000.

[5] B. Fortz and M. Thorup. Increasing internet capacity using local search. *Computational Optimization and Applications*, 29(1):13–48, October 2004.

[6] P. Gajowniczek, M. Pióro, A. Szentesi, and J. Harmatos. Solving an OSPF routing problem with simulated allocation. In *Proc. 1st Polish-German Teletraffic Symposium, PGTS'2000*, 2000. Dresden, Germany.

[7] K. Holmberg and D. Yuan. Optimization of Internet protocol network design and routing. *Networks*, 43(1):39–53, 2004.

[8] ILOG. *CPLEX*. http://www.ilog.com/products/cplex/.

[9] P. Karas and M. Pióro. Optimisation problems related to the assignment of administrative weights in the IP networks' routing protocols. In *Proceedings of the 1st Polish-German Teletraffic Symposium PGTS'2000*, pages 185–192, September 2000.

[10] S. Koehler and A. Binzenhoefer. MPLS traffic engineering in OSPF networks - a combined approach. In *Proceedings of ITC 18*, August – September 2003.

[11] J. Moy. OSPF version 2. *Internet RFC 2328*, April 1998.

[12] E. Mulyana and U. Killat. A hybrid genetic algorithm approach for OSPF weight setting problem. In *Proc. 2nd Polish-German Teletraffic Symposium, PGTS'2002*, pages 39–46, September 2002. Gdansk, Poland.

[13] E. Mulyana and U. Killat. Optimization of IP networks in various hybrid IGP/MPLS routing schemes. In *Proc. 3rd Polish-German Teletraffic Symposium, PGTS'2004*, pages 295–304, September 2004. Dresden, Germany.

[14] R. Networks N. Shen and H. Smit. Calculating igp routes over traffic engineering tunnels. *Network Working Group Internet Draft*, June 2004.

[15] K. Nichols, V. Jacobson, and L. Zhang. A scalable and hybrid IP network traffic engineering approach. *Network Working Group Internet Draft*, December 2001.

[16] D. Papagiannaki, N. Taft, Z. Zhang, and C. Diot. Long-term forecasting of Internet backbone traffic: Observations and initial models. In *Proc. of IEEE INFOCOM 2003*.

[17] H. Pham and B. Lavery. Hybrid routing for scalable IP/MPLS traffic engineering. In *Proceedings of IEEE International Conference on Communications*, volume 1, pages 332–337, 2003.

[18] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[19] M. Pióro, A. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski. On OSPF related network optimization problems. *Perf. Evaluation*, 48:201–223, 2002.

[20] A. Riedl. A hybrid genetic algorithm for routing optimization in ip networks utilizing bandwidth and delay metrics. In *Proc. of IEEE Workshop on IP Operations and Management (IPOM)*, 2002. Dallas, USA.

[21] A. Riedl. Optimized routing adaptation in IP networks utilizing OSPF and MPLS. In *Proceedings of IEEE ICC*, May 2003.

[22] E. Rosen, A. Visiwanathan, and A. R. Callon. Multiprotocol label switching architecture. *Internet RFC 3031*, January 2001.

[23] A. Tomaszewski, M. Pióro, M. Dzida, M. Mycek, and M. Zagożdżon. Valid inequalities for a shortest-path routing optimization problem. In *Proceedings of the 3rd International Network Optimization Conference INOC'2007, Spa*, 2007.

[24] A. Tomaszewski, M. Pióro, M. Dzida, and M. Zagożdżon. Optimal shortest path routing for Internet networks. In *Proceedings INOC'2005, Lisbon*, volume B2, pages 393–400, March 2005.

[25] Y. Zhanga, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM SIGMETRICS*, 2003. USA.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 177 -188

# Flow control in connection oriented communication networks with unisochronic feedback and non-persistent sources

PRZEMYSŁAW IGNACIUK [a] ANDRZEJ BARTOSZEWICZ [b]

[a] Ericpol Telecom
9a Targowa St., 90-042 Łódź, Poland
*przemyslaw.ignaciuk@ericpol.pl*

[b] Institute of Automatic Control
Technical University of Łódź
18/22 Stefanowskiego St., 90-924 Łódź, Poland
*andpbart@p.lodz.pl*

**Abstract:** In this paper the congestion control problem in a connection-oriented communication network is addressed. In the considered mutli-source network the feedback information is provided by means of control units generated by each source once every $M$ data packets. Since the sources adjust the transmission rate only at the control unit arrival, the interval between successive rate modifications is input dependent and varies with time. A new, nonlinear strategy effectively combining the Smith principle with proportional controller with saturation is proposed. Conditions for data loss elimination and full bottleneck link bandwidth utilization are formulated and strictly proved. The presented strategy allows for full resource usage even though the sources are not persistent. Moreover, since the controller saturation limits are fully adjustable, the algorithm performance may be easily tuned according to the existing system resources.

**Keywords:** congestion control, connection-oriented networks, sampled data systems

## 1. Introduction

In recent years, the rapid expansion of long distance network traffic has trigged intensive research aimed at improving efficiency of data transmission [1]–[8]. A valuable survey of earlier congestion control mechanisms is given in [4]. Lengliz and Kamoun [6] introduced a proportional plus derivative controller which is computationally efficient and can be easily implemented in connection-oriented networks. Imer *et al.* [3] gave a brief, excellent tutorial exposition of the congestion control problem and presented new stochastic and deterministic control algorithms. Adaptive control strategies for data flow regulation have been proposed by Laberteaux *et al.* [5]. Their strategies reduce convergence time and improve queue length management.

Due to the significant propagation delays which are critical for the closed loop performance, several researchers applied the Smith principle to control the flow of data in communication networks [1, 2, 7, 8]. In [7], Mascolo considered a single connection

congestion control problem in a general packet switching network. He used the deterministic fluid model approximation of packet flow and applied transfer functions to describe the network dynamics. In the next paper [8], the same author applied the Smith principle to the network supporting multiple connections with different propagation delays. The proposed control algorithm guarantees no data loss, full network utilization, and ensures exponential convergence of queue levels to stationary values without oscillations or overshoots. On the other hand, in paper [1] linear, discrete-time flow control strategy for connection-oriented networks has been proposed. The strategy combines the Smith principle with the discrete time proportional controller. A nonlinear algorithm exploiting the idea of the Smith prediction for the flow regulation was suggested in [2]. The described continuous time control mechanism guarantees congestion alleviating features and full resource usage even though the propagation delays in the multi-source network can be determined only with limited degree of accuracy.

In this paper, the flow control in a connection-oriented communication network is considered. Our approach is similar to that introduced in [1, 2, 7, 8], however as opposed to [1, 7, 8], we propose a nonlinear control strategy. Moreover, in contrast to [2, 7, 8], where continuous time control schemes where elaborated and [1], where discrete-time controller with constant sampling period was proposed, in this paper, an algorithm, which explicitly takes into account irregularities in the feedback information availability, is considered. The strategy combines the Smith principle with the proportional controller with saturation. Our approach guarantees full bottleneck node link utilization and no data loss in the network. As a result, the need of packet retransmission is eliminated and the maximum throughput is achieved. The main novelty of the proposed strategy, as compared with the earlier results based on the Smith's principle, is that it allows for entire bandwidth usage at the bottleneck link even though the sources do not always obey the rate adjustment command (due to temporal or inherent transfer limitations) and may deliver less data than determined by the controller.

## 2. Network model

The telecommunication network considered in this paper consists of data sources, intermediate nodes connected via bidirectional links and destinations. The data from each source traverses a number of switches, which operate in the store-and-forward mode without the traffic prioritization, to be finally delivered to its destination. However, somewhere on the transmission path a node is encountered, whose output link cannot handle the incoming flow. Consequently, congestion occurs and packets, which constitute the data stream, accumulate in the buffer allocated for that link.

In a general case, $n$ data flows pass through the bottleneck node and its output connection and, hence, participate in the flow regulation process. The feedback mechanism for the input rate adaptation is provided by means of the control units sent by each source every $M$ data packets. These special units travel along the same path as the data packets. However, unlike the data packets, they are not stored in the queues at the

intermediate nodes. Instead, once they appear at the switch input link and the feedback information is incorporated, they are immediately transferred at the appropriate output port. As soon as the control units reach destination, they are turned back to be retrieved at the origin and to be used for the transfer speed adjustment round trip time after they were generated. The input rate adaptation at the instant of the management unit arrival only is justified by the fact that this is a specific moment, when meaningful information about the current system condition is delivered to a source. Since the control units are generated every $M$ ordinary packets, the time period between the arrivals of the consecutive management units depends on the emission rate round trip time earlier, which, in turn, changes according to the variations of the network state and makes the feedback unisochronic.

The presented scenario is illustrated in Fig. 1. Source $j$ ($j$ = 1, 2,..., $n$) sends data at the rate $a_j(t)$, where $t$ denotes time. After forward propagation delay $T_{fj}$ packets belonging to the $j$-th virtual connection reach the bottleneck node and are served according to the bandwidth availability at the output link. The remaining data accumulates in the queue. The controller placed at the bottleneck switch compares the current queue length, which at the time $t$ will be denoted as $x(t)$, with its demand value $x_d$, and calculates the aggregate transmission speed $\tilde{a}(t)$. The $n$-th share of the total rate is recorded as the feedback information in every management unit passing through the node. Once the control units from source $j$ appear at the respective end system, they are turned back to arrive at their origin backward propagation delay $T_{bj}$ after being processed by the switch. Sampling module (SM) extracts the flow rate from control units and adjust the transfer speed of the source. Since control units are not subject to queuing delays, round trip time $RTT_j = T_{fj} + T_{bj}$ remains constant for the duration of the connection for each flow.

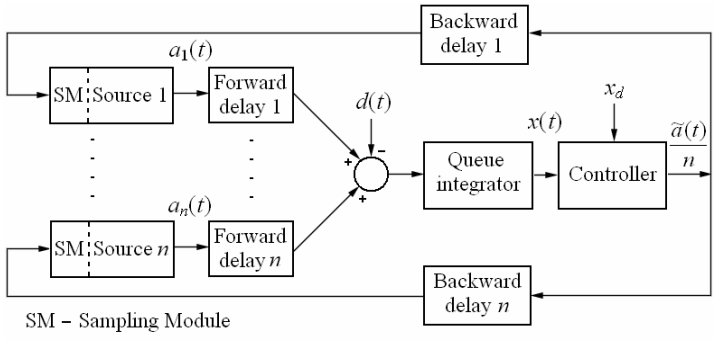



Fig. 1. Network model.

The available bandwidth is modeled as an a priori unknown, bounded function of time $d(t)$. It is lower-bounded by a positive real constant $d_{\min}$ and limited from above by the maximum value $d_{\max}$, i.e. $0 < d_{\min} < d(t) < d_{\max}$. Notice that this definition of the available bandwidth is quite general and it accounts for any standard distribution typically analyzed in the considered problem. If there are packets ready for the transmission in the buffer, then the bandwidth actually consumed by all the sources $h(t)$ will be equal to the available

bandwidth. Otherwise, the output link is underutilized and the exploited bandwidth matches the data arrival rate at the node. Thus, we may write

$$0 \le h(t) \le d(t) \le d_{\max}. \tag{1}$$

The rate of change of the queue length at any instant of time depends on the data arrival speed and on the buffer depletion rate, which, in turn, is directly related to the consumed bandwidth $h(t)$. Therefore, for any $t \ge 0$ the length of the queue at the node may be expressed as

$$x(t) = \sum_{j=1}^{n} \int_0^t a_j(\tau - T_{fj}) d\tau - \int_0^t h(\tau) d\tau. \tag{2}$$

Let us denote by $b_j(t)$ the rate calculated by the controller and sent for the $j$-th source at the instant of the management unit passing through the node. Once the control unit arrives back at the source, the input rate is adjusted to the value determined by the bottleneck node. However, the source cannot always obey the command due to inherent or temporal transfer limitations. Assuming that the sources begin transmission at the time $t = 0$, the $j$-th source rate $a_j(t)$ can be expressed as follows

$$\forall_j \forall_{t<0} a_j(t) = 0 \text{ and } \forall_j \forall_{t\ge 0} a_j(t) = f_j(t) b_j(t - T_{bj}), \tag{3}$$

where $f_j(t)$, representing the source transfer limitations, is subject to the constraint

$$0 < f_{\min} \le f_j(t) \le 1. \tag{4}$$

Since the signal $b_j(t)$ constitutes a vital part of the proposed control scheme, its proper definition will be given together with the description of the flow regulation strategy in the subsequent section.

As a consequence of (3), no packets arrive at the congested node before $T_{f\min} = \min_{j=1,2,\ldots,n} (T_{fj})$. Suppose that initially (before the time $t = 0$) the bottleneck buffer was empty, i.e. $x(t < 0) = 0$. Then, for any time instant smaller than or equal to $T_{f\min}$ the queue length is zero, i.e. $x(t \le T_{f\min}) = 0$.

Let us denote by $t_{j,k}$ the $k$-th moment of time ($k = 1, 2,...$) when the control unit belonging to the $j$-th virtual connection data flow arrives back at source $j$. We assume that the first packet transferred by each source is the control unit so that the information about the current network state could be received at the data origin as quickly as possible. As the sources begin transmission at the time instant $t = 0$, then for $k = 1$ we have $t_{j,1} = RTT_j$. Furthermore, since control units are sent every $M$ data packets, $t_{j,k+1}$ can be determined from the following relation

$$\int_{t_{j,k}}^{t_{j,k+1}} a_j(\tau - RTT_j) d\tau = M. \tag{5}$$

Definition (5) makes sense only for the nonnegative rates $a_j(t)$. Clearly, any control algorithm should be constructed in such a way that this condition is satisfied for every

signal $a_j(t)$. Moreover, although the sources cannot always transmit data at the rate established by the controller, we assume that $a_j(t)$ cannot be lower than the minimum rate $a_{\min}/n$ in order to provide basic responsiveness to the changes of networking conditions. Hence, each source emits a control unit (and after *RTT* adjusts the transfer speed) at least every $T_C = Mn/a_{\min}$.

## 3. Control algorithm

Modern telecommunication systems demand efficient resource exploitation. Therefore, a successful control strategy should guarantee that

(i) data is not lost due to congestion,

(ii) there are always some data packets ready for the transmission in the buffer so that the available bandwidth at the output link of the bottleneck switch is entirely utilized.

In the sequel we propose a new, nonlinear flow regulation algorithm, which fulfills these goals.

The rate calculated by the controller $\tilde{a}(t)$ is determined from the relation given below

$$\tilde{a}(t) = \begin{cases} a_{\min}, & \text{if } W(t) < a_{\min} \\ W(t), & \text{if } a_{\min} \le W(t) \le a_{\max} \\ a_{\max}, & \text{if } W(t) > a_{\max} \end{cases} \tag{6}$$

where $a_{\min}$ and $a_{\max}$ denote the lower and upper limit of the possible flow rate values and $0 < a_{\min} < a_{\max}$. We define the function $W(t)$ as

$$W(t) = K\left[x_d - x(t) - S(t)\right], \tag{7}$$

where $K > 0$ is the controller gain and $x_d > 0$ is the demand queue length. The component

$$S(t) = \sum_{j=1}^{n} \int_{t-RTT_j}^{t} b_j(\tau)\, d\tau \tag{8}$$

is responsible for the Smith prediction and it represents the 'in-flight' data. The rate $b_j(t)$, sent and recorded by the switch for source $j$ at the instant of the control unit passing through the node, is determined from the following equation

$$b_j(t) = \begin{cases} 0, & \text{for } t < -T_{bj} \\ a_{\min}/n, & \text{for } -T_{bj} \le t < T_{fj} \\ \tilde{a}(t_{j,k} - T_{bj})/n, & \text{for } t \ge T_{fj} \text{ and } t \in \left[t_{j,k} - T_{bj}, t_{j,k+1} - T_{bj}\right) \end{cases} \tag{9}$$

Therefore, source $j$ sends data to the network at the minimum rate $a_{\min}/n$ in the $[0, RTT_j)$ interval. Afterwards, it delivers the packets at the rate determined by the controller (according to (7) and (9)) at the discrete time instants $t_{j,k} - T_b$.

We assume that in order for the network to be able to transport the user data at least at the minimum rate, $a_{\min} \le d_{\min}$. On the other hand, to make it possible to always exploit all of the available bandwidth, we expect $a_{\max} \ge d_{\max}$. Therefore, since the utilized bandwidth is limited from below by either the available bandwidth or the incoming flow rate (when the queue length is zero), $h(t)$ will satisfy the following inequalities

$$a_{\min} \le d_{\min} \le h(t) \le d_{\max} \le a_{\max}. \tag{10}$$

To fulfill the first design objective (i), it suffices to show that the queue length never increases beyond a given limit irrespective of the bandwidth changes. Then, if we assign the buffer capacity at least equal to the indicated maximum, no packet will be discarded as a result of congestion, and the risk of losing data will be eliminated.

**Theorem 1:** If sources transmit data according to the conditions formulated by (3)-(9), then for any $t \ge 0$ the queue length at the bottleneck node is upper-bounded by $q_{\max}$, where

$$q_{\max} = x_d - a_{\min}\left(\lambda + 1/K\right) + \left(a_{\max} - a_{\min}\right)T_C, \tag{11}$$

provided that

$$x_d > a_{\max}\left(\lambda + 1/K\right), \tag{12}$$

where $\lambda = \sum_{j=1}^{n} RTT_j / n$ denotes the mean round trip time for the considered set of $n$ connections.

**Proof:** The rate is adjusted by each source at the discrete time instants $t_{j,k}$, and the effect of the modification affects the total arrival rate at the bottleneck node after forward propagation delay. Let us denote by $\theta_m$ ($m$ = 1, 2,...) the moments of time when the total arrival rate changes as a consequence of the transfer speed adjustment at some source. The first such modification coincides with the arrival of the initial packet belonging to the flow with the shortest forward delay, so for $m$ = 1 we have $\theta_1 = T_{f\min}$. Since each source updates the transmission rate at least every $T_C$, the interval

$$\alpha_m = \theta_{m+1} - \theta_m \tag{13}$$

between any two consecutive potential changes of the total incoming rate at the congested switch is subject to the constraint $0 \le \alpha_m \le T_C$. The interval $\alpha_m = 0$ reflects the case, when the modification of the transmission speed, which occurred at two or more sources, influences the aggregate rate at the switch at the same moment of time.

Let us denote the queue length at the time instant $\theta_m$ by $x_m = x(\theta_m)$. For $m$ = 1 and $\theta_1 = T_{f\min}$ we can write $x_1 = x(\theta_1) = 0 < q_{\max}$. Therefore, the proposition holds for any moment of time $t \le T_{f\min}$.

Let us consider some $m > 1$ and the queue length at a time instant $t \in [\theta_m, \theta_{m+1})$

$$x(t) = x_m + P(\theta_m, \theta_m + \delta) - \int_{\theta_m}^{\theta_m+\delta} h(\tau)d\tau, \tag{14}$$

where $t = \theta_m + \delta$ and $\delta \in [0, \alpha_m)$. The function

$$P(\tau_1, \tau_2) = \sum_{j=1}^{n} \int_{\tau_1}^{\tau_2} a_j(\tau - T_{fj}) d\tau \tag{15}$$

represents the amount of data that arrived at the bottleneck node between $\tau_1$ and $\tau_2$ time instants.

In order to analyze the queue length variations in the time interval $[\theta_m, \theta_{m+1})$, we examine the behavior of function $W$. We will consider two cases: first, the situation when $W(\theta_m) > a_{\min}$, and, afterwards, the circumstances when $W(\theta_m) \le a_{\min}$.

**Case 1:** We analyze the situation when $W(\theta_m) > a_{\min}$. From the definition of function $W$, we get

$$x_m < x_d - a_{\min} / K - S(\theta_m). \tag{16}$$

Assumption (12) guarantees that expression on the right-hand side of inequality (16) is positive. Applying (16) to (14), we arrive at

$$x(t) < x_d - a_{\min} / K - S(\theta_m) + P(\theta_m, \theta_m + \delta) - \int_{\theta_m}^{\theta_m + \delta} h(\tau) d\tau. \tag{17}$$

The biggest amount of data arrives at the node if all the sources are able to follow the controller command, i.e. if $\forall_j f_j(t) = 1$. On the other hand, the quantity on the right hand side of (17) is the biggest in the situation when $\delta > RTT_j$. Then, the number of incoming packets exceeds the predicted one. Consequently, since the individual source rate is upper-bounded by $a_{\max}/n$, the difference $P(\theta_m, \theta_m + \delta) - S(\theta_m)$ can be evaluated as

$$P(\theta_m, \theta_m + \delta) - S(\theta_m) \le a_{\max} (\delta - \lambda). \tag{18}$$

Using (10) and (18), we get the following estimate of the queue length (17)

$$\begin{aligned} x(t) &< x_d - a_{\min} / K + a_{\max} (\delta - \lambda) - a_{\min} \delta = \\ &= x_d - a_{\min} (\lambda + 1/K) + (a_{\max} - a_{\min})(\delta - \lambda). \end{aligned} \tag{19}$$

Since $\delta \le T_C$,

$$x(t) < x_d - a_{\min} (\lambda + 1/K) + (a_{\max} - a_{\min})(T_C - \lambda) \le q_{\max}. \tag{20}$$

This ends the first part of the proof.

**Case 2:** Now, let us examine the situation when $W(\theta_m) \le a_{\min}$. We begin with finding the last moment $t^* < \theta_m$ when the signal $W(t)$ was greater than $a_{\min}$. It should be stressed at this point, that since the control unit emission at the sources and rate generation at the congested node are not synchronized, $t^*$ does not have to coincide with any of the $\theta_m$ time instants. According to (9) and (12), $W(-T_{b\max})$, where $T_{b\max} = \max_{j=1,2,\ldots,n} (T_{bj})$, satisfies the following

$$W(-T_{b\max}) = K\left(x_d - 0 - 0\right) > a_{\max} > a_{\min}. \tag{21}$$

For $t \in (-T_{b\max}, T_{f\min})$ signal $W$ decreases with time and, due to assumption (12), at the upper margin of the indicated interval attains the value greater than $a_{\min}$, i.e. $W(T_{f\min}) > a_{\min}$. This means that moment $t^*$ actually exists and $t^* > T_{f\min}$. The value of $W(t^*)$ satisfies the following inequality

$$W(t^*) = K\left[x_d - x(t^*) - S(t^*)\right] > a_{\min}. \tag{22}$$

After the term rearrangement, we obtain

$$x(t^*) < x_d - a_{\min} / K - S(t^*). \tag{23}$$

The queue length at a time instant $t \in [\theta_m, \theta_{m+1})$ can be expressed as

$$x(t) = x(t^*) + P(t^*, t) - \int_{t^*}^{t} h(\tau)d\tau, \tag{24}$$

where $t = \theta_m + \delta$, $\delta \in [0, \alpha_m)$ and $\alpha_m$ is defined by (13). Applying (23) to (24), we get

$$x(t) < x_d - a_{\min} / K - S(t^*) + P(t^*, t) - \int_{t^*}^{t} h(\tau)d\tau. \tag{25}$$

The term $P(t^*, t) - S(t^*)$ describes the difference between the number of data units which arrived at the bottleneck node during the interval from $t^*$ to $t$, and the number of packets still 'in flight' at the time moment $t^*$, i.e. those for which the sending rate has already been calculated and which have not yet arrived at the buffer of the bottleneck node. The biggest number of packets will arrive if the sources have no limitations, i.e. if $\forall_j f_j(t) = 1$. Then, $a_j(t) = b_j(t - T_{bj})$, and from definitions (8) and (15) we obtain

$$P(t^*, t) - S(t^*) = \sum_{j=1}^{n} \int_{t^*}^{t} b_j(\tau)d\tau - S(t). \tag{26}$$

The time instant $t^*$ is the last moment when the algorithm computed a value greater than $a_{\min}$. Afterwards, the established flow rate was equal to $a_{\min}$, so difference (26) can be evaluated as

$$\sum_{j=1}^{n} \int_{t^*}^{t} b_j(\tau)d\tau - S(t) \le a_{\max}T_C + a_{\min}\left(t - t^* - T_C\right) - a_{\min}\lambda. \tag{27}$$

Recall that $h(t \ge T_{f\min}) \ge a_{\min}$. Then, using (27) we get the following estimate of the queue length (25)

$$\begin{aligned} x(t) &< x_d - a_{\min} / K + a_{\max}T_C + a_{\min}\left(t - t^* - T_C\right) - a_{\min}\lambda - a_{\min}\left(t - t^*\right) = \\ &= x_d - a_{\min}\left(\lambda - 1/K\right) + \left(a_{\max} - a_{\min}\right)T_C = q_{\max}. \end{aligned} \tag{28}$$

This concludes the proof.

Full link utilization, as formulated by (ii), requires the presence of data packets in the bottleneck node buffer at any instant of time. In other words, if $x(t)$ is greater than zero, then all of the available bandwidth at the congested link is consumed. The theorem presented below shows how the demand queue length $x_d$ should be selected so that this condition is fulfilled.

**Theorem 2:** If sources transmit data according to the conditions formulated by (3)-(9), $f_{\min}a_{\max} > d_{\max}$ and the demand value of the queue length satisfies the following inequality

$$x_d > a_{\max}\left(\lambda + 1/K\right) + \left(f_{\min}a_{\max} - a_{\min}\right)T_C, \tag{29}$$

then for any $t > T_{f\,\min} + T_C + T_{\max}$, where $T_{\max} = q_{\max} / (f_{\min}a_{\max} - d_{\max})$, the queue length is strictly positive.

**Proof:** The theorem assumption implies that we deal with the time instants $t > T_{f\,\min} + T_C + T_{\max} > \theta_1$. Considering some $m > 1$ and the value of signal $W$ at the moment of the node input rate modification $\theta_m$, we may distinguish two cases: the situation when $W(\theta_m) < a_{\max}$, and the circumstances when $W(\theta_m) \geq a_{\max}$.

**Case 1:** We consider the situation when $W(\theta_m) < a_{\max}$. Directly from the definition of function $W$, we obtain

$$x_m > x_d - a_{\max} / K - S(\theta_m). \tag{30}$$

Let us examine the queue length at some time instant $t \in [\theta_m, \theta_{m+1})$ as was defined in (14). Using (30), we get

$$x(t) > x_d - a_{\max} / K - S(\theta_m) + P(\theta_m, \theta_m + \delta) - \int_{\theta_m}^{\theta_m+\delta} h(\tau)d\tau. \tag{31}$$

The Smith predictor term $S(t) \leq a_{\max}\lambda$. On the other hand, each source transmits packets at least at the $a_{\min}/n$ rate, so $P(\theta_m, \theta_m + \delta) \geq a_{\min}\delta$. Therefore, we can state the following

$$x(t) > x_d - a_{\max} / K - a_{\max}\lambda + a_{\min}\delta - \int_{\theta_m}^{\theta_m+\delta} h(\tau)d\tau. \tag{32}$$

Since the utilized bandwidth is upper-bounded by $d_{\max} < f_{\min}a_{\max}$, and $\delta \leq T_C$, the queue length expressed by (32) will satisfy the condition given below

$$x(t) > x_d - a_{\max} / K - a_{\max}\lambda + a_{\min}\delta - d_{\max}\delta > 0. \tag{33}$$

This concludes the first part of the proof.

**Case 2:** Now, let us investigate the situation when $W(\theta_m) \geq a_{\max}$. First, notice that according to (29) the assumptions of Theorem 1 are fulfilled. We seek for the last moment $t^* < \theta_m$ when signal $W$ was smaller than $a_{\max}$. It comes from Theorem 1 that the queue length never exceeds $q_{\max}$. At the same time, the packet depletion rate is limited by $d_{\max}$. On the other hand, in the situation when the assigned rate equals $a_{\max}$, the incoming rate at the node (after taking into account the transfer limitations of the sources) will not

be lower than $f_{\min}a_{\max}$. Thus, in order to preserve the buffer space indicated in Theorem 1, the controller may continuously set the biggest rate $a_{\max}$ for the maximum period $T_{\max} = q_{\max} / (f_{\min}a_{\max} - d_{\max})$, and the instant $t^*$ actually exists. Since $t^*$ was the last instant, when $W < a_{\max}$, and the time separation between any two consecutive aggregate input rate modifications does not exceed $T_C$, $t^* \geq \theta_m - (T_{\max} + T_C)$. The value of $W(t^*)$ satisfies the inequality given below

$$W(t^*) = K\left[x_d - x(t^*) - S(t^*)\right] < a_{\max}. \tag{34}$$

Rearranging terms in (34), we get

$$x(t^*) > x_d - a_{\max} / K - S(t^*). \tag{35}$$

The queue length at some moment of time $t \in [\theta_m, \theta_{m+1})$ may be expressed as in (24). Applying (35) to (24), we obtain

$$x(t) > x_d - a_{\max} / K - S(t^*) + P(t^*,t) - \int_{t^*}^{t} h(\tau)d\tau. \tag{36}$$

According to (6), $S(t^*) \leq a_{\max}\lambda$. Recall that $t^*$ was the last instant before $t$, when the controller calculated rate smaller than $a_{\max}$. This rate can be as low as $a_{\min}$. Afterwards, the algorithm computed the maximum value of the flow rate. Since the control units appear at the discrete time instants, the rate assignment can be delayed, but not more than by $T_C$. Moreover, due to the source transfer capability limitations, the incoming rate at the node (although predicted to be at the maximum) may be as low as $f_{\min}a_{\max}$. Thus, the amount of the incoming data $P$ will satisfy the following

$$P(t^*,t) \geq a_{\min}T_C + f_{\min}a_{\max}\left(t^* - t - T_C\right). \tag{37}$$

For any $t$ the utilized bandwidth $h(t) \leq d_{\max}$, so substituting (37) into (36), we arrive at

$$x(t) > x_d - a_{\max} / K - a_{\max}\lambda + a_{\min}T_C + f_{\min}a_{\max}\left(t^* - t - T_C\right) - d_{\max}\left(t - t^*\right). \tag{38}$$

The theorem assumption implies $f_{\min}a_{\max} > d_{\max}$. Since $t > t^*$,

$$x(t) > x_d - a_{\max}\left(\lambda + 1/K\right) - \left(f_{\min}a_{\max} - a_{\min}\right)T_C > 0. \tag{39}$$

This ends the proof.

## 4. Simulation results

In order to verify the control strategy proposed in this paper, simulation tests were performed using Matlab-Simulink®. First, the model of a wide area network with irregular period of feedback information availability was constructed according to the description provided in Section 2. Three connections ($n = 3$), characterized by the delays: $RTT_1 = 20$ ms, $RTT_2 = 30$ ms and $RTT_3 = 70$ ms, participate in the flow regulation process. Each source sends control units every $M = 32$ equal-size data pieces. The maximum available bandwidth $d_{\max}$ was set as 9100 packets/s, the lower bound of the

overall source rate $a_{\min}$ as 960 packets/s, and the upper bound $a_{\max}$ as $18300 > d_{\max}/f_{\min} = 9100/0.5$ packets/s. The controller gain $K$ was adjusted to 20 $s^{-1}$ and the demand queue length $x_d$, calculated according to (29), was set as $2470 > 2466$ packets. The bandwidth actually available for the controlled connections $d(t)$ is illustrated in Fig. 2 by curve (b), and the rate determined by the controller is shown as curve (a). We can see from the graph that function $d$ experiences sudden changes of large amplitude, which reflects the most rigorous networking conditions. The queue length $x(t)$ resulting from applying the proposed regulation scheme is shown in Fig. 3. As we can see, the queue length never exceeds the value of 4115 packets and does not drop to 0. These two properties imply no buffer overflow and full bottleneck link utilization. The rates assigned for the sources (curve a) and the true transfer speeds of each transmitter (curve b) are shown in Fig. 4. It is clear from the plots that the sources cannot always deliver data at the rate established by the controller (the actual rate can be as low as half of the assigned one). Despite significant rate limitations experienced by the transmitters, the proposed strategy guarantees the maximum throughput in the network.

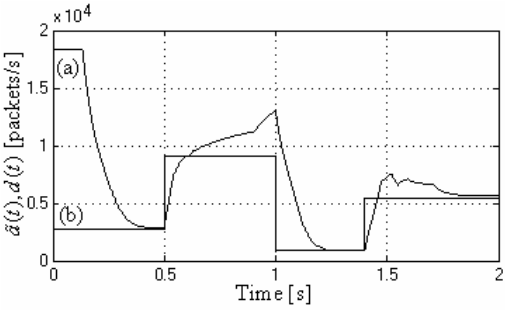



Fig. 2. Total rate (a) and available bandwidth (b).

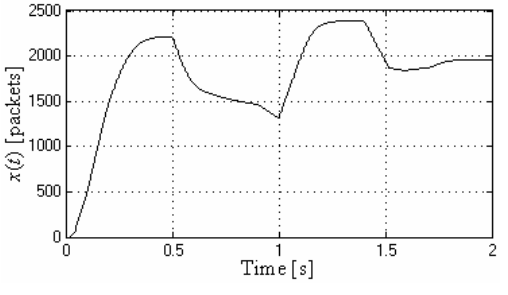



Fig. 3. Queue length.

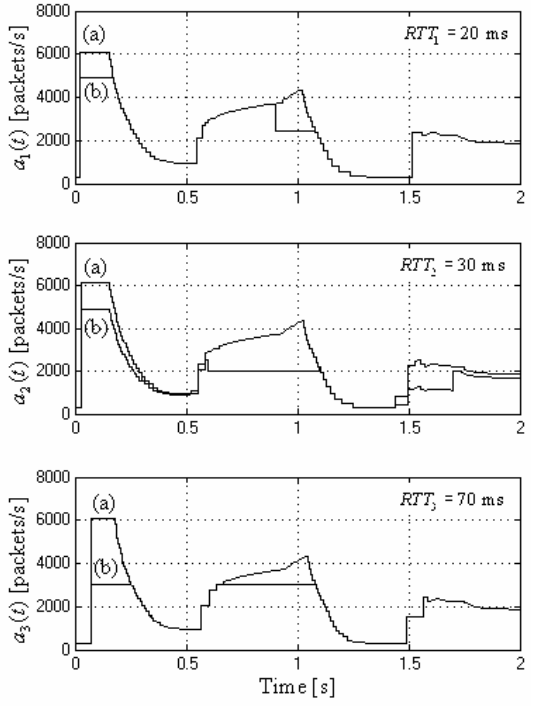



Fig. 4. Source rates: (a) – assigned, (b) – real.

## 4. Conclusions

In this paper the congestion control problem in a connection-oriented network supporting multiple data flows was addressed. The proposed nonlinear algorithm effectively combines the Smith principle with the proportional controller with saturation. Conditions for no packet loss and full network utilization were presented and strictly proved with explicit consideration of the irregularities in the feedback information availability. As the transfer speed limits are fully adjustable within the bandwidth constraints, the algorithm performance may be easily tuned by the user. Moreover, since the rates generated by the controller are always nonnegative and bounded, the proposed strategy may be directly implemented in real systems. The main advantage of the algorithm proposed in this paper over previously published Smith's predictor based solutions is that it allows for total resource usage despite possible transfer limitations of the sources. Since low rates may result from the transmitter itself, or may be caused by congestion occurring elsewhere in the network (and not at the node where the controller operates) the proposed algorithm may efficiently coexist with other, possibly different flow regulation schemes in the communication system.

## Acknowledgment

This work has been supported by the Polish State Committee for Scientific Research (KBN) grant no. 3 T10A 037 28 (2005-2007).

## References

[1] A. Bartoszewicz and T. Molik, "ABR traffic control over multi-source single-bottleneck ATM networks," *Journal of Applied Mathematics and Computer Science*, 14(1), pp. 43-51, 2004.

[2] A. Bartoszewicz.: *Nonlinear flow control strategies for connection-oriented communication networks*. Proceedings of the IEE – Part D Control Theory and Applications, 153(1), pp. 21-28, 2006.

[3] O. C. Imer, S. Compans, T. Basar, and R. Srikant.: *Available bit rate congestion control in ATM networks*. IEEE Control Systems Magazine, 21(1), pp. 38-56, 2001.

[4] R. Jain.: *Congestion control and traffic management in ATM networks: recent advances and a survey*. Computer Networks and ISDN Systems, 28(13), pp. 1723-1738, 1996.

[5] K. P. Laberteaux, Ch. E. Rohrs, and P. J. Antsaklis.: *A practical controller for explicit rate congestion control*. IEEE Transactions on Automatic Control, 47(6), pp. 960-978, 2002.

[6] I. Lengliz and F. Kamoun.: *A rate-based flow control method for ABR service in ATM networks*. Computer Networks, 34(1), pp. 129-138, 2000.

[7] S. Mascolo.: *Congestion control in high-speed communication networks using the Smith principle*. Automatica, 35(12), pp. 1921-1935, 1999.

[8] S. Mascolo.: *Smith's principle for congestion control in high-speed data networks*. IEEE Transactions on Automatic Control, 45(2), pp. 358-364, 2000.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 189–199

# Techno-economic Challenges in Interconnection between Network Operators

MIROSŁAW KANTOR, PIOTR CHOŁDA, JAN DERKACZ, ANDRZEJ JAJSZCZYK [a]
ANGEL O. FERREIRO [b]

[a]Department of Telecommunications, AGH University of Science and Technology
30-059 Kraków, Poland
Telephone: (+48 12) 617–28–52, Fax: (+48 12) 634–23–72
{kantor,cholda,derkacz,jajszczyk}@kt.agh.edu.pl

[b]Telefonica I+D
Emilio Vargas 6, 28043-Madrid
olivo@tid.es

**Abstract:** A Least Cost Routing (LCR) solution supporting optimal routing decisions in the inter-carrier environment is proposed. The solution could play an important role for operators which face different routing options with regard to service quality and cost. The issue is presented in the context of the optimal traffic partner-carrier choice. The optimization problem is formulated. Especially, different tariff formulations are taken into account. Then, an example scenario and an optimal solution for the considered problem are elaborated.
**Keywords:** interconnection, optimization, routing, tariff

## 1. Introduction

As competition in the telecommunications industry continues to intensify, carriers and service providers seek ways to increase revenues and reduce operating costs (OPEX). Due to the development of Next Generation Networks, leading to a multi-service transport layer within a multi-domain environment, the importance of interconnection issues keeps growing. Also the number of Service Providers (like ISPs) is increasing rapidly and the structure of the industry goes towards an unbundled value chain, where large diversity of services can be offered by many telecommunication market players. Thus, the new business models combined with the net neutrality requirement [1] enforces the network operators to change interconnection policy used until now. To support network services, carriers build and diversify interconnect partner relationships [2]. Network service providers often negotiate special interconnection agreements and tariffs. Thus, they can provide the best service to their end-customers [3]. However, as network services become connectionless and network connectivity increases, operators face numerous routing alternatives.

On the other hand, changes in tariff plans, which are complex and may use highly granular pricing models, should be considered as a characteristic feature of this market. These characteristics and the explosive growth in voice and data traffic could lead carriers to deploy new call routing models if the business environment becomes very dynamic and routing changes are required in shorter time frames. Moreover, operators require the ability to ensure that calls in their networks are routed according to the lowest cost route to maximize their income. Inefficiencies in implementing interconnection strategies can decrease carriers' outcome, as well as make them waste time.

Therefore, the need for developing algorithms supporting the choice of optimal interconnection routes becomes crucial. In this paper, we introduce the concept of the Least Cost Routing (LCR) solution, which helps to optimize connections between telecommunication operators by minimizing costs for served demands and maximizing an efficient use of the existing network infrastructure. By using LCR algorithms, the routing strategy can be more efficiently executed by incorporating the knowledge of cost and margin with network conditions. The LCR algorithm can also shorten time needed to analyze a huge number of alternatives and help a carrier make decisions considering new agreements with other carriers within a dynamic framework.

## 2. Problem Description

In a multi-domain environment, possibly combining incumbent and virtual network operators, service providers and operators who want to send traffic to a certain destination have many possibilities to choose routes through such domains. Since an important element of an operation budget is the interconnection cost, the critical problem is to route traffic through the cheapest routes in order to control OPEX. However at the same time, the operator has to take into consideration the requirements regarding the quality of service (QoS), resilience as well as bandwidth issues. The chosen route has to meet all the requirements with reference to both price and quality. Such a route is called the optimal one.

To help the operators find the best routes, the Least Cost Routing (LCR) solution has been proposed. General LCR model is presented in Fig. 1. In order to calculate the optimal routes the implemented algorithms take into account multiple network-based parameters like operators' price lists, network configuration, traffic history, etc. The most important parameter for choosing the routes is the cost of transiting/terminating connections (given in tariffs as the price per time/volume unit). However, the quality of service of the offered interconnection in a general case can be also taken into account when calculating the optimal route. The latter issue is not considered in this paper.

Based on a series of reference data the optimization program as well as the heuristic algorithm implemented in LCR find automatically the optimal route configurations for all interconnect traffic, assigning the most appropriate Points of Interconnection (POI) for specific traffic directions and termination points. The considered model is the static one, i.e, the traffic distribution is proposed in a long time period, e.g., one month. Traffic is assumed to

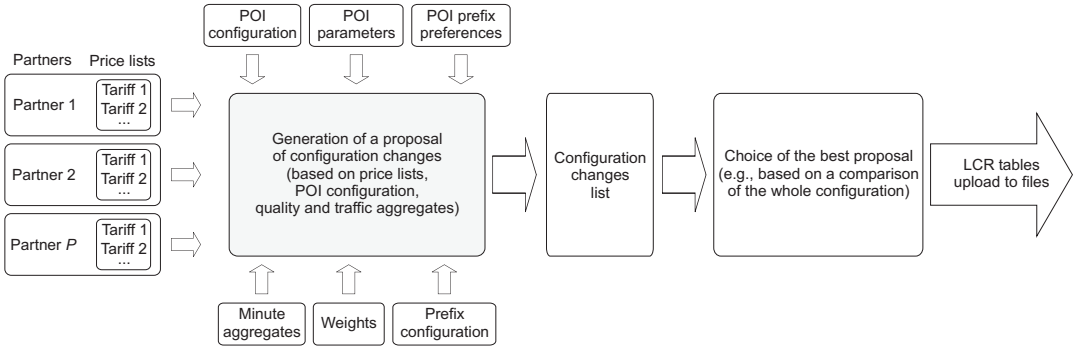



Fig. 1. General LCR model (POI: Point of Interconnection)

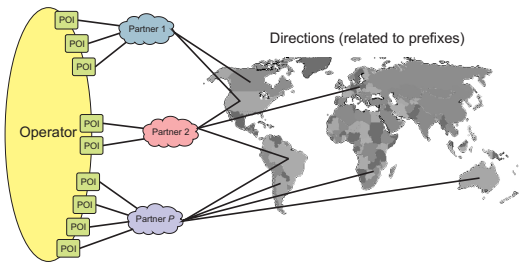
reach a maximum value, considering possible peaks and a given network availability, during that period. LCR functionalities give the operator a network configuration table that can be manually, after an administrator's approval, or automatically implemented into the network control systems. The delivered solution is based on a general connection model shown in Fig. 2(a). This connection model is presented from a point of view of an operator who wants to send the traffic to specific destinations/prefixes. It is assumed that the operator has agreements with a certain number of partners who offer the possibility of transiting or terminating the traffic for some directions. The interconnection partners not always offer the connections to all destinations. In the case of big operators they offer the global services indeed, but smaller ones operate rather in narrower geographical zones.

The operator usually has many points in its network through which the traffic is exchanged with partners. These points are known as POIs (Point of Interconnection). Formally, POI can be defined as geographical location where two networks interconnect and exchange traffic. It is possible to define some POIs with the same interconnection partner (see Fig. 2(b)). In the considered case, the developed algorithm does not differentiate between locations of POIs. It is assumed that the location of the POIs are given.

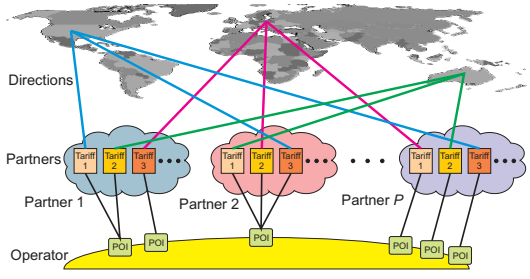
The parameters related to the POI are the tariffs and capacity (Fig. 2(c)). Different tariffs can be specified in the same POI, even on the same direction. It is due to the fact that interconnection partner can offer different classes of service (e.g., gold class, bronze class, etc.) on the same directions/prefixes with different prices. The capacity at the POI, i.e., the maximum traffic volume which can be sent through this POI is another important factor. This limitation can result from technical constraints of particular locations (e.g., the link capacity). The capacity of a POI is also considered as a given value and in the case of our algorithm it is a constraint.

The main problem to be solved by the LCR algorithm is to propose the optimal distribution of the routes for the whole traffic outgoing from an operator. Therefore, the volume of the traffic to be carried to different directions (see $H_1$, $H_2$ and $H_3$ in Fig. 2(d)) is the main input for the developed algorithm.

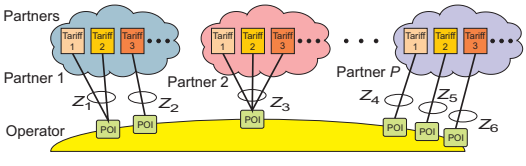
Configured weights for particular routes are also considered (see Fig. 2(e)). These



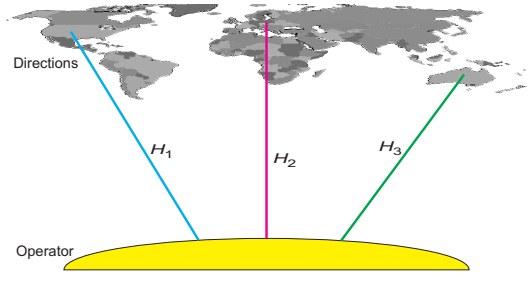

(a) General connection model



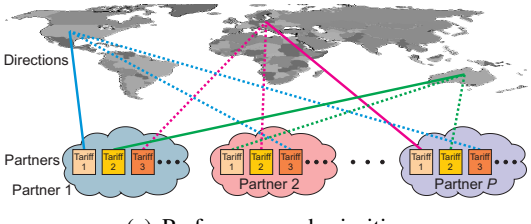

(b) General POI model




(c) POI parameters: tariffs and interface capacity constraints




(d) Volume of traffic to be carried to different directions




(e) Preferences and priorities

Fig. 2. LCR model

weights can result from operator's corporate priorities and preferences. The priority means that, independently of the fact that sending traffic through the particular route or to the particular destination is more expensive in comparison with other possible options, the traffic is transmitted just through that partner. This situation occurs in the telecommunications interconnection market, as the operator enter into a privileged agreement with a particular partner. The 'preferences' term means that in the case of routes with the same cost offered by different partners the service delivered by a preferred partner is chosen.

The developed algorithm takes also into consideration rules based on the date, day of a week, and hour of a day. However, the considered model is the static one, i.e., the traffic distribution is proposed in the monthly perspective. Nevertheless, this model, after taking into account some more parameters and variables can be easily adapted to the dynamic envi-

ronment.

We also have to remember that we need credible statistical data describing the total traffic related to the period of time for which the planning is made. Therefore, the use of traffic forecasting techniques is encouraged. Nevertheless, in this paper, we assume that such statistics is provided.

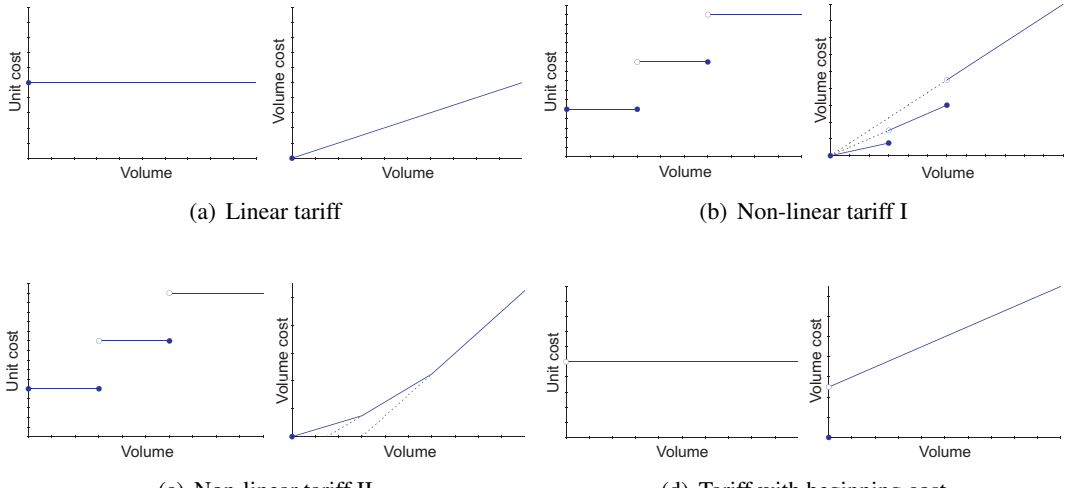
## 3. Cost Functions

The proposed model concerns the most popular tariffs. As a tariff we understand a scheme of rates and regulations managing charging of telecommunication services. A tariff model is in most cases composed of the following two elements [4]:

- a price: a monetary component;
- a related tariff model, i.e., a calculation scheme, which provides a charging function enabling calculation of costs considering charging variables (e.g., time of usage, volume transferred, allocated bandwidth) and charging coefficients (e.g., price per suitable unit).

Numerous tariff models have been proposed for telecommunication services [5]. These models can be classified into three groups [4]: linear tariff models, non-linear tariff models and discounts.

In the case of linear tariffs the price per defined unit (volume, time) of usage is equal for all units. The total cost of a call or transmission is simply calculated by multiplication of the unit price by the number of carried units (Fig. 3(a)).




(a) Linear tariff (b) Non-linear tariff I

(c) Non-linear tariff II (d) Tariff with beginning cost

Fig. 3. Cost structures of different tariffs (unit and total cost)

The non-linear tariffs [6] result in different prices per used unit (Fig. 3(c)). The marginal cost changes when some thresholds are exceeded. The new cost is related only to the amount of minutes which exceed the thresholds. The first non-linear tariff, sometimes called the total volume traffic tariff (Fig. 3(b)), is characterized with the marginal cost changes when some thresholds are exceeded. For the whole minute volume carried in such a tariff only one unit cost is used at once, i.e., the cost related to the highest exceeded threshold. The most popular and simplest non-linear tariff is the flat rate tariff [7]. Under a flat pricing scheme the user is charged a fixed amount per time, irrespective of usage. Further tariff models can be defined by mixing linear and non-linear tariff models to obtain more sophisticated tariff models. An example of such a combined model is the two-part tariff model which consists of a fixed entrance fee for a certain period of time and a per-unit charge (Fig. 3(d)).

Two-part interconnection tariffs exist in a number of ways [8]: many interconnection tariffs consis of a call setup charge, plus a per minute charge. Many interconnection arrangements require initial or ongoing payments for the dedicated interconnection capacity (both traffic and signalling) between networks, plus the on-going usage related charges. Now, most operators charge their clients for the number of minutes routed to other operators. However, there are many different ways of setting up a contract between operators. Many contracts can be described as a general function of the number of minutes. One possible approximation is 'pay per minutes' where the price is a linear function of the number of minutes served with the option of limiting the total number of minutes. The cost of a connected call can be computed in several different ways. One of them is a weighted combination of the quality of service and price. To differentiate their portfolios, the carriers also introduce so called time-bands, i.e., different time units, for which prices are defined. Those different time intervals are known as the peak-on and peak-off intervals with different prices [9]. Discounts (called here global promotions) are a special type of tariff model applied to decrease the total cost for the customer. They can be defined over total costs incurred for a special type of transaction or on total costs incurred for a certain period of time. Discounts are applied in combination with tariff models and result in an additional reduction of prices.

## 4. Optimization Heuristic

The formulation of the LCR problem is based on Mixed-Integer Programming (MIP). The mathematical model of the considered problem has been described in details in [10]. As for a large number of variables this problem is unsolvable, we propose a heuristic algorithm to find the best solution (see Fig. 4). The heuristic solution is computed as follows:

1. In the first step, the initial data for the algorithm, such as the POIs configuration, connection scheme and traffic distribution are determined.
2. Based on the data from step one, the potential partners and interfaces for required destinations are chosen. The list of the partners and interfaces can be ordered according to some criteria, e.g., the capacity of interface, the traffic limits, the lowest price, the highest quality or the mix of the previous criteria.

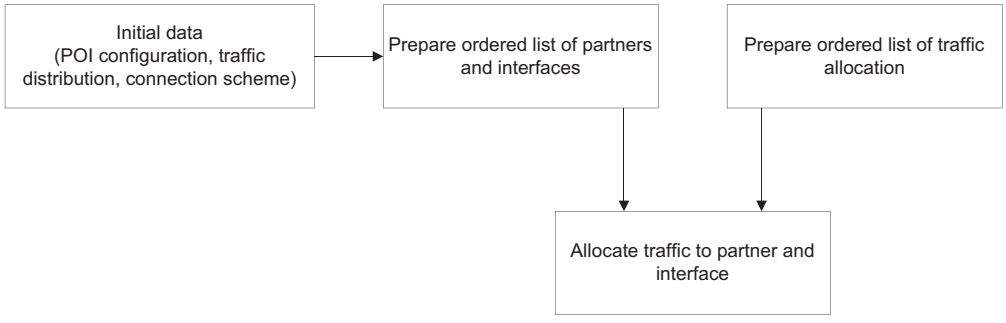


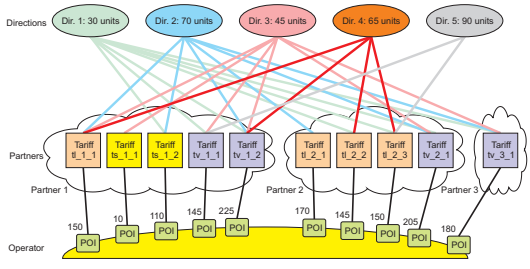

Fig. 4. The proposed heuristic algorithm

3. Simultaneously, the ordered list of traffic allocation is prepared. The list consists of the destinations, where the traffic has to be transmitted. The traffic can be ordered according to some rules, e.g., from the highest to the lowest value, according to some preferences, etc.
4. In the last step, the choice of the partner and the interface which meets the required criteria is made. The traffic granularity defined as the number of minutes taken into account in one heuristic loop, as well as the step of heuristic, meant as the duration of time for which the traffic distribution for destinations is looked for:
   (a) to start the heuristic the granularity and step is determined;
   (b) at first, the traffic to priority destinations is allocated;
   (c) in the next step, the rest of the traffic is allocated according to the chosen rule, e.g., the cheapest path is chosen;
   (d) if the limit of the traffic for a partner or interface is exceeded for the just serviced destination, the previous traffic allocation is canceled except the traffic allocation to priority destinations; the considered destination is assumed to be the priority destination and is served as the first from among the non-priority destinations.

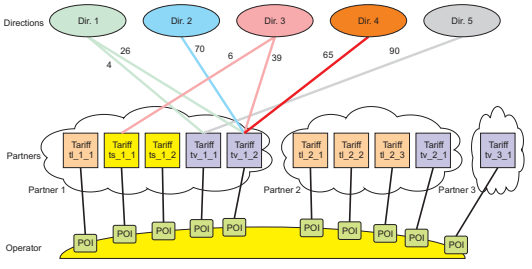
## 5. Numerical Results

To show the LCR solution performance, a scenario is specified by a given number of interconnection partners, possible routes, interfaces constraints, tariff definitions and volume traffic requirements (see Fig. 5(a)).

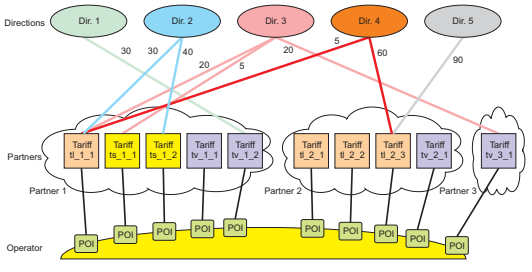
We assume that the operator wants to send: 30 traffic units on direction 1, 70 traffic units on direction 2, 45 traffic units on direction 3, 65 traffic units on direction 4, 90 traffic units on direction 5. In the presented example, we consider $P = 3$ interconnection partners and $D = 5$ directions/prefixes. Partners 1 and 2 can send the traffic to all the destinations while




(a) Constraints




(b) Results: optimal goal function is 755.3




(c) Results: results based on the heuristic

Fig. 5. Numerical example

partner 3 only to direction 1, 2 and 3. The capacities of the POIs are also shown, e.g., 150 units limiting POI connected to tariff $tl_{1_1}$. Interconnection partners offer different number of tariffs: partner 1 five tariffs, partner 2 four, and partner 3 only one, respectively. The structure of all used tariffs is described in Table 1. The linear (see Fig. 3(a)) and non-linear tariff structure are used (see Fig. 3(c)). The linear tariffs are as follows: $tl_{1_1}$, $tl_{2_1}$, $tl_{2_2}$, $tl_{2_3}$; the non-linear — $ts_{1_1}$, $ts_{1_2}$, $tv_{1_1}$, $tv_{1_2}$, $tv_{2_1}$, $tv_{3_1}$. The global promotion has been applied by partners 1 and 2, while partner 3 does not offer any discount (see Table 2). The cost used in the tariffs definition is given in some monetary units.

The goal of the optimization algorithm implemented in the LCR solution is to find the distribution of the traffic which has to be sent on 5 directions in the cheapest way, i.e., to

Table 1. Numerical example: tariffs structures

| $tl_{1_1}$ | | $ts_{1_1}$ | | $ts_{1_2}$ | | $tv_{1_1}$ | | $ts_{1_2}$ | | $tl_{2_1}$ | | $tl_{2_2}$ | | $tl_{2_3}$ | | $tv_{2_1}$ | | $tv_{3_1}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $c_1$ | 1 | $c_1$ | 6 | $c_1$ | 5 | $c_1$ | 3 | | | | | | | $c_1$ | 2 | $c_1$ | 3 |
| | | $t_1$ | 3 | $t_1$ | 40 | $t_1$ | 70 | $t_1$ | 150 | | | | | | | $t_1$ | 180 | $t_1$ | 20 |
| $c$ | 8 | $c_2$ | 2 | $c_2$ | 9 | $c_2$ | 9 | $c_2$ | 4 | $c$ | 6 | $c$ | 8 | $c$ | 5 | $c_2$ | 9 | $c_2$ | 4 |
| | | $t_2$ | 6 | $t_2$ | 100 | $t_2$ | 120 | $t_2$ | 200 | | | | | | | $t_2$ | 195 | $t_2$ | 140 |
| | | $c_3$ | 5 | $c_3$ | 10 | $c_3$ | 10 | $c_3$ | 8 | | | | | | | $c_3$ | 10 | $c_3$ | 9 |

$c$ means: unit cost, $t$ means: threshold value

Table 2. Numerical example: tariffs structures

| **Partner** | 1 | 2 | 3 |
|---|---|---|---|
| **Threshold** | 250 | 200 | No discount |
| **Discount** | 30% | 20% | |

choose the POI, partners and tariffs in order to pay as small as possible. The optimal traffic load is presented in Fig. 5(b). To calculate the exact solution we use the commercial MIP solver CPLEX [11]. In the considered scenario the optimal goal function is 755.3. According to the results based on the exact optimization algorithm only partner 1 has been chosen for transiting/terminating the traffic. If the results of the algorithm are to indicate the potential partner for agreement, in case of such a structure of data (traffic, tariffs, interfaces constraints) as in the example, it is advisable to make an agreement only with partner 1.

On the other hand, Fig. 5(c) presents results obtained on the basis of the proposed heuristic algorithm. We can see that, contrary to the exact solution, the heuristic algorithms eagerly allocated linear tariffs. The value of the goal function in this case is 1512. On the basis of this sub-optimal value with the exact optimum, we imply that the heuristics must be made much more sophisticated.

## 6. Operator Considerations

Theoretical assumptions and models need to be verified and modified from the viewpoint of real conditions in which operates an existing telecommunications service provider. In particular, regulatory and physical constraints need to be considered. Mutual long term bilateral agreements established between operators should be taken into account as well. For instance, the interconnection is strongly regulated by the Government Regulatory Authority in Spain as well as in many other countries. Soon, such a regulation is going to be applied at the EU level. The interconnection must be supported by the predominant network operator (an old monopoly holder) and the tariffs calculated on the basis of cost and the initial investment credit. It is assumed that the situation must evolve in the sense of loosing restrictions towards a liberalized market where discounts and bilateral agreements will naturally emerge. However, for the moment, flat tariffs are predominant in a wholesale scheme and the two-part interconnection tariffs are applied to mutual interconnection agreements, mainly for the mobile access segment where timebands can be defined in the three different schemes:

- Working days: 8:00 till 16:00 is the expensive timeband for business.
- Working days: 18:00 till 22:00 is another expensive timeband for residential users.
- Weekends: There is no expensive timeband for the access segment.

As far as the interconnection for wideband services (e.g. Internet access) is included separately in the group of a flat tariff, there is no place for non linear interconnection agree-

ments. In fact, this kind of interconnection agreements is most common for long-distance and wholesale services, where other considerations aside from cost estimation are used, especially in trans-national links (e.g., in Southern America).

## 7. Conclusions

In the paper, we present the approach to the least-cost routing in the inter-carrier context. First, the problem is defined. Especially, we overview the typical tariffs used in the inter-carrier operations.

The further work will aim at developing the heuristics for the considered problem. Fast computation of the given scenario by CPLEX resulted from not too sophisticated tariffs definitions, small traffic to transmit, etc. In real-size problems, the time for achieving the optimal solution could be too long. We plan to compare the optimal results with sub-optimal solutions obtained one the basis of a heuristic. Quality of Service issues will also be considered in the further algorithm. The static model presented in this paper will be extended to the dynamic case.

## Acknowledgement

This work has been done within the EU FP6 NoE e-Photon/ONe+ (`http://www.e-photon-one.org`) and NOBEL2 (`http://www.ist-nobel2.org`) frameworks.

This work was supported by the Polish Ministry of Science and Higher Education under Grant N517 013 32/2131.

The authors want to thank Sofie Verbrugge for her help while writing this paper.

## References

[1] J. Crowcroft: Net Neutrality: The Technical Side of the Debate: A White Paper, *ACM SIGCOMM Comp. Comm. Rev.*, vol. 37, pp. 49-55, Jan. 2007.

[2] M. Weiss, S. Shin: Internet Interconnection Economic Model and its Analysis: Peering and Settlement, *Proc. 17$^{th}$ World Computer Congress*, Montreal, Canada, August 2002.

[3] C. Courcoubetis, R. Weber: *Pricing Communication Networks*, New York, John Wiley & Sons, Inc., 2003.

[4] K. Stanoevska-Slabeva: Tariff Models for Telecommunication Services in a Liberalised Market, *The Int. J. Media Man.*, vol. 3, no. 1, pp. 33-38, 2001.

[5] B. Mitchell, I. Vogelsang: *Telecommunications Pricing*, Cambridge, Cambridge University Press, 1991.

[6] J.-H. Hahn: Nonlinear Pricing of Telecommunications with Call Network Externalities, *Int. J. of Ind. Org.*, vol. 21, pp. 949-967, Sept. 2003.

[7] L. McKnight, J. Bailey: Internet Economics: When Constituencies Collide in Cyberspace, *IEEE Intern. Comp.*, Nov./Dec. 1997.

[8] J. Sandbach: Two-part Interconnection Tariffs, *Proc. ITS European Conference*, Porto, Portugal, Sept. 4-6, 2005.

[9] Malta Communications Authority: *Fixed Interconnection Pricing Review*, Malta, 2006. [Online]. `http://www.mca.org.mt`

[10] P. Chołda, M. Kantor, A. Jajszczyk, K. Wajda: Least Cost Routing in Inter-Carrier Context, *Proc. Globecom 2006*, 27 November - 1 December 2006, San Francisco, USA. [Online].
`http://www.kt.agh.edu.pl/~cholda/Papers/GlobeCom2006.pdf`

[11] ILOG S.A.: *ILOG CPLEX 9.1. User's Manual*, Gentilly Cedex, France, Apr. 2005.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 201 -212

# The impact of scheduler on the maximum admissible load to a class of service

ROBERT JANOWSKI

Institute of Telecommunications
Warsaw University of Technology
*rjanowsk@tele.pw.edu.pl*

**Abstract:** In this paper we propose the methods for analysing the most popular scheduling algorithms like Priority Queueing (PQ), Generalized Processor Sharing (GPS) and Priority Queueing - Generalized Processor Sharing (PQ-GPS), which handle the traffic in two or more classes of service. Based on this analysis we determine the maximum admissible load which guarantees the assumed IP packet loss ratio (IPLR) for each class of service. We compare our results with a well-known and widely applied decomposition method. The decomposition method doesn't take into account the specific features of the scheduling algorithm but relies on decomposing a complex system into a number of separate single queue sub-systems each corresponding to one classes of service. The comparison between our method and the decomposition one, proves that by taking into account the specific features of the schedulers, we benefit from higher values of the maximum admissible load which means higher utilization of the network resources.

**Keywords:** Quality of Service, Admissible load, Schedulers, PQ, GPS, PQ-GPS.

## 1. Introduction – statement of the problem

The concept of classes of service (CoS) has been already accepted as one of the approaches for assuring a target level of quality of service (QoS) in IP (*Internet Protocol*) networks [8] expressed in terms of IP packet transfer delay (IPTD), IP packet delay variation (IPDV) or IP packet loss ratio (IPLR) [9], [10] parameters. This concept assumes the usage of different schedulers and possibly the separate, per CoS buffers for handling the packets streams which differ in traffic characteristics and QoS requirements essentially. Additionally, to assure target values of IPTD, IPDV and IPLR parameters the load of each CoS must be limited to some threshold value controlled by the enforcement of the Call Admission Control (CAC) function.

Existing methods for calculating this load limit (also referred as the maximum admissible load or AC limit) rely on the decomposition of the systems with complex schedulers into a number of "sub-systems" each representing one CoS [12], [13]. The crucial point of the decomposition method is the way of partitioning the link capacity among the supported CoSs. Usually, it is so called "static partitioning" which assigns each CoS a fraction of the link capacity i.e. a constant bandwidth independent of the traffic conditions in other CoSs. The amount of assigned capacity is either implied by the

minimum guaranteed capacity as in case of GPS like schedulers or is chosen arbitrarily as in case of the highest priority in case of PQ schedulers. For instance, following the guidelines of the decomposition method the PQ-GPS scheduler with 3 CoSs would be represented as a set of 3 independent single queue sub-systems with dedicated buffers and capacity (see

Fig. 1).

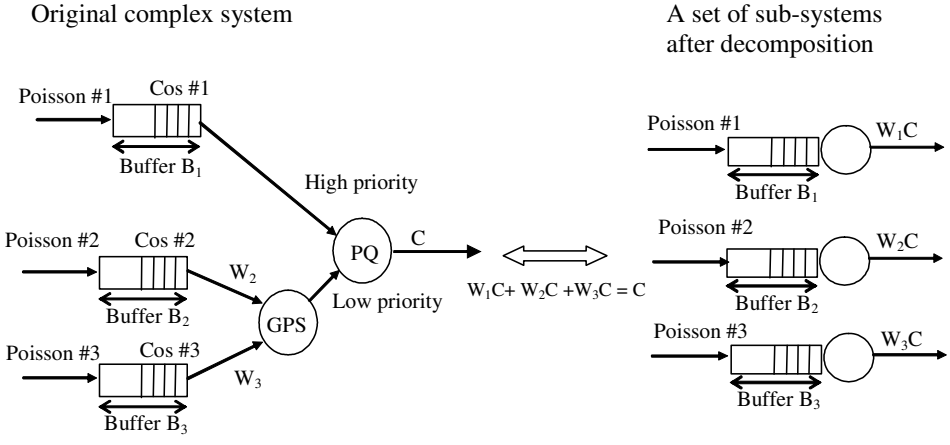



Fig. 1 The principles of the decomposition method

Dividing the link capacity among CoSs is needed in order to treat each CoS separately while calculating its maximum admissible load (or equivalently AC limit). Then the determination of the maximum admissible load of particular CoS becomes much easier since now we deal with a single server queue for which we can apply a variety of well known methods.

While determining the maximum admissible load for a CoS we assume that the value of *IPDV* is controlled by appropriately dimensioned buffer size i.e. short enough to guarantee that the maximum queuing delay is less than IPDV. Accordingly, the maximum admissible load is limited by requirements on the IPLR value.

In order to relate the admissible load to the buffer size B and the target value of *IPLR*, first we determine the tail of the probability distribution of the queue size. For obtaining the tail of the queue size probability distribution we use Generating Function approach and the approximation by the dominating pole outlined in [11] and [18]. Briefly sketching the approach, it consists of the following steps:

- determine the generating function of the number of the packets in the system (the system state – N(z)); for single server queue it is well known Pollaczek-Khintchine formulae [22] – see (1),

$$N(z) = (1-\rho)\frac{A(z)(z-1)}{z-A(z)} \tag{1}$$

- approximate the probability distribution of the number of packets in the system by geometric distribution with decaying rate determined by the dominating pole $z_0$ (i.e. such a value of z which is the real and closest to 1 solution of the function from the denominator of (1) ),

$$\Pr ob\{N = n\} \approx K_0 (\frac{1}{z_0})^n. \tag{2}$$

Afterwards, we assume that *IPLR* is well approximated by the buffer overflow probability and the constant $K_0$ is close to 1, what implies the following simple relationship between the *IPLR* value and the system parameters: buffer size B and the value of the dominating pole $z_0$:

$$IPLR \approx (\frac{1}{z_0})^B. \tag{3}$$

From (3) it is apparent that requirements for the target value of IPLR while the buffer size B is fixed, implies the particular value of $z_0$.Since the dominating pole $z_0$ is found by solving the equation from the denominator of (1), in case of Poisson input stream, it is determined from:

$$z_0 - e^{\rho(z_0 - 1)} = 0 \tag{4}$$

By combining the equations (3) and (4) we can obtain final formulae determining the maximum admissible load $\rho_{max}$ which assures target IPLR value on the buffer size B. For the i-th CoS which is dedicated the buffer size $B_i$ and for which the packet losses not greater than $IPLR_i$ are required, this formulae is as follows:

$$\rho_{\max,i} = \frac{Ln(\frac{1}{IPLR_i})^{\frac{1}{B_i}}}{(\frac{1}{IPLR_i})^{\frac{1}{B_i}} - 1}. \tag{5}$$

The proposed formulae is relatively simple and so it is easy to apply. However, by statically partitioning the whole link capacity C between CoSs before calculating the admissible load, we ignore the important features of scheduling algorithm. These features include mutual impact of the traffic handled by different CoSs: utilizing the whole capacity left over by low priority traffic in case of PQ scheduler or the ability to assure the minimum capacity with the possibility to consume more if any other CoS is under-utilized in case of GPS scheduler.

## 2. Analysis of the most popular scheduling algorithms

In this section we provide the analysis of the most popular scheduling algorithms that are implemented in the legacy equipment. Regarding the analyzed system we assume that the packets arriving to each CoS form Poisson stream. This assumption is relevant when the traffic is generated by a large number N of flows and the buffer sizes do not scale

with N (which is our case) [20]. Otherwise, when the scaling occurs, the more appropriate model of the input traffic is Gaussian process. Furthermore, the packets are constant size specific for the given CoS. Since we consider at most 3 CoSs, the packet sizes follow the relations: $d_2=d_3$, $d_2/d_1=d$ and $d_3/d_1=d$. Each CoS is dedicated a separate buffer of the size $B_1$, $B_2$ and $B_3$ for CoS#1, CoS#2 and CoS#3, respectively.

### 2.1 Priority Queueing (PQ)

For PQ system we only consider two CoSs: CoS#1 handled with higher priority and CoS#2 handled with lower priority. It means that the analyzed PQ system is a sub-system of the one depicted in Fig. 1 consisting of only CoS#1 and CoS#2 exactly as they are denoted there.

The analysis of PQ system with Poisson input and generally distributed packet sizes has been already carried out in [7]. There, the analysis assumed two priority levels and provided the results in terms of generating function of the joint probability distribution of both queues. Despite this ready to use result which could be adopted for our purposes, we developed our own analysis. It differs from [7] by explicitly carrying out the analysis from the point of view of CoS#2 handled with lower priority. In our analysis the impact of high priority traffic on the low priority one is modelled by introducing unavailability periods when server is busy with high priority packets. Essentially our results agree with the results obtained in [7] while differing only with respect to some constants. This is due to the fact that we inspect the queue size just before completing the service of a packet while in [7] the joint queue size is inspected at the end of each time unit (a slot) corresponding to the service time of a shorter packet.

For our analysis we differentiate among the following 5 cases:

- previous observation took place at the end of an empty slot and during the unavailability period no low priority packet has arrived at the system; next observation takes place 1 slot after the unavailability period (Fig. 1-A),
- previous observation took place at the end of an empty slot and during the unavailability period at least one low priority packet has arrived at the system; next observation takes place d slots after the unavailability period (Fig. 1-B),
- previous observation took place at the end of service completion of the last low priority packet and during the unavailability period no low priority packet has arrived at the system; next observation takes place 1 slot after the unavailability period (Fig. 1-C),
- previous observation took place at the end service completion of the last low priority packet and during the unavailability period at least one low priority packet has arrived at the system; next observation takes place d slots after the unavailability period (Fig. 1-D),
- previous observation took place at the end of service completion of the low priority packet and there were at least two low priority packets in the system; next observation takes place d slots after the unavailability period (Fig. 1-D).

We introduce the following notation where $N_2(n)$ denotes random variable describing the system state at the n-th observation, $A_{1,i}(n+1)$ (respectively $A_{2,i}(n+1)$) denotes random variable describing the number of high (respectively low) priority packets arriving at the

system during a single i-th slot occurring after n-th observation and by U(n) the length of the unavailability period (counted in slots) occurring after n-th observation.

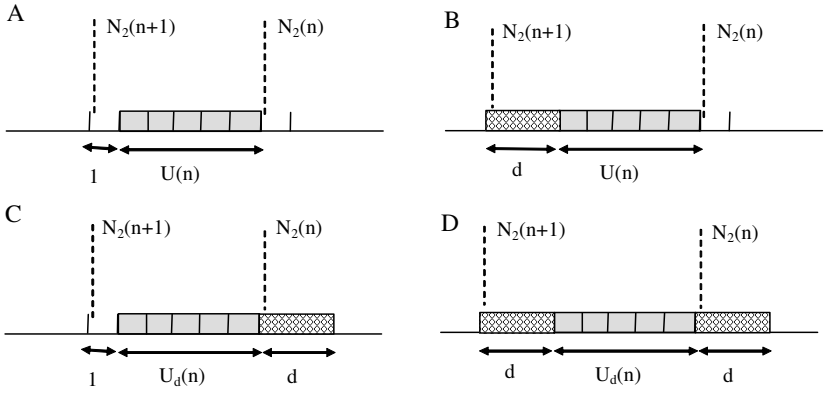



Fig. 1 All possible cases of the system state transitions between consecutive observations

We summarize the transitions between system states in consecutive observations as defined above in the following equation (each component presented in a separate line corresponds to one of the 5 cases defined above):

$$\begin{aligned} N_2(z) &= N_2(0)A_2(0)U(A_2(0)) + \\ &+ N_2(0)A_2^{d-1}(z)(A_2(z)U^d(A_2(z)) - A_2(0)U(A_2(0))) + \\ &+ \Pr ob\{N_2 = 1\}A_2(0)U^d(A_2(0)) + \\ &+ \Pr ob\{N_2 = 1\}A_2^{d-1}(z)(A_2(z)U^d(A_2(z)) - A_2(0)U^d(A_2(0))) + \\ &+ \left(\frac{N_2(z)}{z} - \frac{z\Pr ob\{N_2 = 1\} + N_2(0)}{z}\right)A_2^d(z)U^d(A_2(z)) \end{aligned} \tag{6}$$

After some algebra, we finally find the following expression for the generating function of the number of low priority packets in the system:

$$N_2(z) = \frac{1-\rho_1-\rho_2}{1-\rho_1-\rho_2+\lambda_2}\,\frac{z(1-A_2^{d-1}(z)) + A_2^d(z)(zU(A_2(z)) - U^d(A_2(z)))}{z - A_2^d(z)U^d(A_2(z))}, \tag{7}$$

where the function U(z) can be determined from the following equation defined on the basis of the busy period analysis of the high priority traffic:

$$U(A_2(z)) = A_1(A_2(z)U(A_2(z))) \tag{8}$$

Combining the equations 7 and 8, and additionally exploiting the fact that both $A_1(z)$ and $A_2(z)$ represent the generating function of the Poisson distributions, we calculated the tail of the queue size probability distribution based on the approximation by the dominating pole [11], [18]. After that, similarly as in section 1 we used the buffer overflow probability in place of the IPLR value and after some algebra we finally found

the following expression for the maximum admissible load for the CoS#2 handled with lower priority:

$$\rho_{2\max} = \lambda_2 d = \frac{\frac{1}{B_2} Ln[\frac{1}{IPLR_2}] - \lambda_1 d(\frac{1}{IPLR_2}^{\frac{1}{d}\frac{1}{B_2}} - 1)}{(\frac{1}{IPLR_2}^{\frac{1}{B_2}} - 1)}. \quad (9)$$

This result takes into account the mutual ratio of the packet sizes *d* handled by different CoSs (by high and low priority) what was not possible in case of the decomposition method.

### 2.2 Generalized Processor Sharing (GPS)

For GPS system we only consider two CoSs. It means that the analyzed GPS system is a sub-system of the one depicted in Fig. 1 consisting of only CoS#2 and CoS#3 exactly as they are denoted there. Based on the assumptions formulated at the beginning of section 2, the packet sizes $d_2$, $d_3$ are equal and weights $w_2$, $w_3$ are used to differentiate between capacities guaranteed for each of these CoSs.

Such a system has been considered in many works assuming different traffic conditions (e.g exponential service times [15] or Long Range Dependent traffic [16]) and different analysis methods (e.g. worst case analysis [14] or exact analysis with the use of complex functions [15]). Some analysis were explicitly targeted on determining the probability distribution of the queue size of particular CoS [3], [4], [5], [6] what is especially useful when determining the maximum admissible load based on the tail of the distribution. However due to the complex nature of the processes used for modelling the system behaviour (e.g. *Markov Modulated Fluid Process* in [4] and [5] or *Phase Type* distributions in [6]) the solutions were not given in a close-form expression but in an algorithmic way, what is much less convenient for practical purposes.

The main part of our analysis focuses on the bandwidth sharing between two CoSs during the periods when one of them is under-utilized (temporarily has no packets to transmit) so the other can benefit from a higher service capacity than the guaranteed value. The mutual impact of CoSs was modelled taking one simplifying assumption. It stated that one of the CoSs (here CoS#2 without loosing the generality of considerations) could use only its own assigned minimum capacity while the other (here CoS#3) could benefit from utilizing additional capacity left over by CoS#2.

The packets departing from CoS#2 provoke the changes in available bandwidth for CoS#3. The impact of traffic departing from CoS#2 on the CoS#3 can be captured by introducing an additional packet stream ($\lambda_2^*$) which interferes with the packet stream $\lambda_3$ in the way shown in the Fig. 2. In order to characterize the additional stream ($\lambda_2^*$) we start from the equation which indicates that the number of departed packets over some long interval T cannot be greater than the number of arrived packets nor the assigned capacity:

$$D_{2,T} = Min[\sum_{i=1}^{T} A_{2,i}, w_2 T] \cdot \tag{10}$$

To simplify further analysis we exploit Chernoff's inequality [21], which states that for every random variable X and any number *a* the following holds:

$$\Pr ob\{X > a\} \le e^{-aLnz} E\{z^X\}. \tag{11}$$

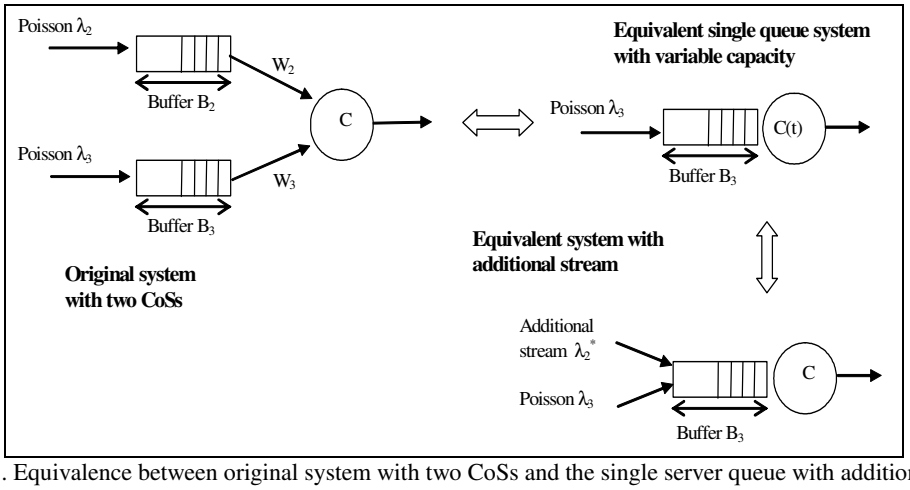



Fig. 2 . Equivalence between original system with two CoSs and the single server queue with additional stream modelling the impact of the other CoS.

Assuming that in (10) the $w_2T$ is not a constraining factor, we have :

$$\Pr ob\{D_{2,T} > a\} \le e^{-aLnz} e^{\lambda_2^* w_2 T(z-1)}. \tag{12}$$

To find the closest upper bound, we need to determine such a *z* that will minimize the value of the exponent in (12). For this purpose we differentiate the exponent and equalize it to 0. We find that the upper bound we look for, is:

$$\Pr ob\{D_{2,T} > a\} \le e^{-aLn[\frac{a}{\lambda_2^* w_2 T}]+a-\lambda_2^* w_2 T}. \tag{13}$$

Using Chernoff's inequality again, but this time for determining the generating function of $D_{2,T}$, we find that :

$$D_{2,T}(z) = E\{z^X\} \ge e^{aLnz} e^{-aLn[\frac{a}{\lambda_2^* w_2 T}]+a-\lambda_2^* w_2 T}. \tag{14}$$

Again, repeating the procedure with finding the closest lower bound, we finally find that the maximum of the exponent in (14) is attained for *a*:

$$a = \begin{cases} \lambda_1^* w_1 T z \quad iff \quad \lambda_1^* w_1 T z < w_1 T \\ w_1 T \quad iff \quad \lambda_1^* w_1 T z \geq w_1 T \end{cases}. \tag{15}$$

Respectively the upper bound for the $D_{2,T}(z)$ is given by:

$$D_{2,T}(z) = \begin{cases} e^{\lambda_2^* w_2 T (z-1)} \quad iff \quad \lambda_2^* z \leq 1 \\ z^{w_2 T} (\lambda_2^*)^{w_2 T} e^{w_2 T (1-\lambda_2^*)} \quad iff \quad \lambda_2^* z > 1 \end{cases}. \tag{16}$$

The first case recalls the situation where the departing packet stream from CoS#2 is not constraint by the assigned minimum capacity $w_2T$ and preserves the original traffic characteristics – it is still Poisson. The second case recalls the situation where the departing packet stream from CoS#2 is really constraint by the minimum capacity $w_2T$. As a result the departing stream is smoothed and its generating function resembles representation of the constant bit rate stream.

To find out the representation of the departing traffic relevant for the one slot time scale (denoted as $D_2(z)$) we postulate that it should be independent and identically distributed in each slot (i.i.d. feature). Thus it is related to $D_{2,T}(z)$ in the following way:

$$D_2(z) = [D_{2,T}(z)]^{\frac{1}{T}} = \begin{cases} [e^{\lambda_2^* w_2 T (z-1)}]^{\frac{1}{T}} = e^{\lambda_2 (z-1)} \quad if \quad \lambda_2 z \leq w_2 \\ [z^{w_2 T} (\lambda_2^*)^{w_{21} T} e^{w_{21} T (1-\lambda_2^*)}]^{\frac{1}{T}} = z^{w_2} (\frac{\lambda_2}{w_2})^{w_2} e^{(w_2 - \lambda_2)} \quad if \quad \lambda_2 z > w_2 \end{cases}. \tag{17}$$

Now, we can proceed with the analysis in the same way as in section 1 for a single server system but with the input traffic being the sum of two packet streams: the original Poisson ($\lambda_3$) and the additional one ($\lambda_2^*$). Accordingly the generating function of the system state for CoS#3 is given by:

$$Q_3(z) = \frac{(1-\rho) A_3(z) D_2(z)(z-1)}{z - A_3(z) D_2(z)}. \tag{18}$$

From the equation (18), applying the approach outlined in section 1, we finally find the following expression for the maximum admissible load for the CoS#3:

$$\rho_{3\max} \equiv \lambda_3 = \begin{cases} \dfrac{\frac{1}{B_3} Ln \frac{1}{IPLR_3} - \lambda_2 (\frac{1}{IPLR_3}^{\frac{1}{B_3}} - 1)}{\frac{1}{IPLR_3}^{\frac{1}{B_3}} - 1} \quad if \quad \lambda_2 \frac{1}{IPLR_3}^{\frac{1}{B_3}} \leq w_2 \\ \dfrac{(1-w_2)\frac{1}{B_3} Ln \frac{1}{IPLR_3} - w_2 Ln \frac{\lambda_2}{w_2} - (w_2 - \lambda_2)}{\frac{1}{IPLR_3}^{\frac{1}{B_3}} - 1} \quad if \quad \lambda_2 \frac{1}{IPLR_3}^{\frac{1}{B_3}} > w_2 \end{cases}. \tag{19}$$

### 2.3 Priority Queueing –Generalized Processor Sharing (PQ-GPS)

The analysis of the whole PQ-GPS system with 3 *CoSs* as depicted in Fig. 1, relies on the results obtained for PQ and GPS systems. In the literature, there are very few references to the methods for calculating the admissible load or even to the methods for analyzing PQ-GPS schedulers. To the best of our knowledge, any contributions about PQ-GPS either rely on simulation studies to characterize the properties of this scheduler

as in [2] or they rely on the decomposition method [1]. Recently in [19] a method for analyzing such a system but with an input traffic exhibiting Long Range Dependence feature was proposed. The literature survey about the analysis of PQ-GPS systems included in [19] confirms our statement about very few contributions in this field.

The final formulae which calculates the maximum admissible load of the CoS#3, was obtained by combining the partial results obtained earlier for PQ and GPS systems:

$$\rho_{3\max} = \lambda_3 d = \begin{cases} \dfrac{\frac{1}{B_3} Ln\frac{1}{IPLR_3} - \lambda_2 d(\frac{1}{IPLR_3}^{\frac{1}{B_3}} - 1) - \lambda_1 d(\frac{1}{IPLR_3}^{\frac{1}{d}\frac{1}{B_3}} - 1)}{(\frac{1}{IPLR_3}^{\frac{1}{B_3}} - 1)} & if \quad \lambda_2 d \frac{1}{IPLR_3}^{\frac{1}{B_3}} \le w_2 . \\ \dfrac{(1-w_2)\frac{1}{B_3} Ln\frac{1}{IPLR_3} - w_2 Ln\frac{\lambda_2 d}{w_2} - (w_2 - \lambda_2 d) - \lambda_1 d(\frac{1}{IPLR_3}^{\frac{1}{d}\frac{1}{B_3}} - 1)}{(\frac{1}{IPLR_3}^{\frac{1}{B_3}} - 1)} & if \quad \lambda_2 d \frac{1}{IPLR_3}^{\frac{1}{B_3}} > w_2 . \end{cases} \tag{20}$$

## 3. Analytical results – comparison with decomposition method

In this section we summarize the analytical results for the maximum admissible load obtained by means of the decomposition method and our proposed method. To better illustrate the benefits from applying the proposed method we have reported the percentage gain indicating the increase in maximum admissible load.

| $B_2$ [pkts] | $\rho_{2max}$ (decomposition method) | $\rho_{2max}$ (proposed method) | | Gain [%] | |
|---|---|---|---|---|---|
| | | $d=d_2/d_1=1$ | $d=d_2/d_1=10$ | $d=d_2/d_1=1$ | $d=d_2/d_1=10$ |
| 10 | 0.486 | 0.486 | 0.544 | 0 | 12 |
| 20 | 0.586 | 0.629 | 0.660 | 7.7 | 12.6 |
| 50 | 0.653 | 0.724 | 0.737 | 11 | 12.9 |
| 100 | 0.676 | 0.758 | 0.764 | 12.1 | 13 |

Tab. 1 PQ system: the maximum admissible load of CoS#2 ($\rho_{2max}$) obtained with proposed method vs $\rho_{2max}$ obtained with the decomposition method. The remaining system parameters are: $IPLR_1=10^{-3}$, $B_1=10$ [pkts], $w_1=0.3$ (resulting in $\rho_1=0.694w_1$), $IPLR_2=10^{-3}$

| $B_3$ [pkts] | $\rho_{3max}$ (decomposition method) | $\rho_{3max}$ (proposed method) | Gain [%] |
|---|---|---|---|
| 10 | 0.347 | 0.377 | 8.6 |
| 20 | 0.419 | 0.490 | 16.9 |
| 50 | 0.466 | 0.585 | 25.5 |
| 100 | 0.483 | 0.619 | 28.2 |

Tab. 2 GPS system: the maximum admissible load of CoS#3 ($\rho_{3max}$) obtained with proposed method vs $\rho_{3max}$ obtained with the decomposition method. The remaining system parameters are: $IPLR_2=10^{-3}$, $B_2=10$ [pkts], $w_2=0.5, w_3=0.5$,( resulting in $\rho_2=0.694w_2$), $IPLR_3=10^{-3}$.

| $B_2$ [pkts] | $\rho_2$ | $B_3$ [pkts] | $\rho_{3max}$ (decomposition method) | $\rho_{3max}$ (proposed method) | | Gain [%] | |
|---|---|---|---|---|---|---|---|
| | | | | d=1 | d=10 | d=1 | d=10 |
| 20 | 0.2511 | 10 | 0.208 | 0.213 | 0.291 | 2.4 | 39.9 |
| 20 | 0.2511 | 20 | 0.251 | 0.319 | 0.360 | 27.1 | 43.4 |
| 20 | 0.2511 | 50 | 0.28 | 0.404 | 0.421 | 44.3 | 50.4 |
| 20 | 0.2511 | 100 | 0.29 | 0.437 | 0.446 | 50.7 | 53.8 |

Tab. 3 PQ-GPS system: the maximum admissible load of CoS#3 ($\rho_{3max}$) obtained with proposed method vs $\rho_{3max}$ obtained with the decomposition method. The remaining system parameters are: $IPLR_1=10^{-3}$, $B_1=10$ [pkts], $w_1=0.4$ (resulting in $\rho_1=0.694w_1$), $IPLR_2=10^{-3}$, $w_2=0.3$, $w_3=0.3$, $IPLR_3=10^{-3}$

To verify the proposed formulas we implemented considered scheduling algorithms under ns-2 simulation platform [17]. In case of GPS and PQ-GPS schedulers the implementation took into account different existing realizations of this general concept of fair bandwidth sharing, like Weighted Fair Queueing with Virtual Time (*WFQ*), Self Clock Fair Queueing (*SCFQ*) or Deficit Round Robin (*DRR*). Simulation results with 95% confidence intervals are summarized in Tab. 4, Tab. 5, Tab. 6 and Tab. 7.

| $B_2$ [pkts] | $IPLR_1$ | | $IPLR_2$ | |
|---|---|---|---|---|
| | $d=d_2/d_1=1$ | $d=d_2/d_1=10$ | $d=d_2/d_1=1$ | $d=d_2/d_1=10$ |
| 10 | $<10^{-6}$ | $7.7\ 10^{-6} \pm 2.3\ 10^{-6}$ | $2.53\ 10^{-4} \pm 1.1\ 10^{-5}$ | $2.89\ 10^{-4} \pm 6.2\ 10^{-5}$ |
| 20 | $<10^{-6}$ | $5.4\ 10^{-6} \pm 5.5\ 10^{-7}$ | $1.71\ 10^{-4} \pm 3\ 10^{-5}$ | $1.61\ 10^{-4} \pm 7.1\ 10^{-6}$ |
| 50 | $<10^{-6}$ | $9.2\ 10^{-6} \pm 1.3\ 10^{-6}$ | $9.5\ 10^{-5} \pm 2\ 10^{-5}$ | $7.7\ 10^{-5} \pm 9.1\ 10^{-6}$ |
| 100 | $<10^{-6}$ | $5.6\ 10^{-6} \pm 1.8\ 10^{-6}$ | $4.3\ 10^{-5} \pm 1.2\ 10^{-5}$ | $3.3\ 10^{-5} \pm 1.5\ 10^{-5}$ |

Tab. 4 Simulated $IPLR_1$ and $IPLR_2$ levels after loading CoS #2 to the values indicated in Tab. 1.

| $B_3$ [pkts] | $\rho_3$ | $IPLR_3$ | | |
|---|---|---|---|---|
| | | WFQ | SCFQ | DRR |
| 10 | 0.377 | $5\ 10^{-4} \pm 4.0\ 10^{-5}$ | $6.8\ 10^{-5} \pm 1.5\ 10^{-5}$ | $7.5\ 10^{-5} \pm 9.5\ 10^{-6}$ |
| 20 | 0.490 | $1.86\ 10^{-4} \pm 2.7\ 10^{-5}$ | $7.0\ 10^{-5} \pm 1.2\ 10^{-5}$ | $7.5\ 10^{-5} \pm 1.3\ 10^{-6}$ |
| 50 | 0.585 | $8.4\ 10^{-5} \pm 2.8\ 10^{-5}$ | $5.1\ 10^{-5} \pm 2.5\ 10^{-5}$ | $8.4\ 10^{-5} \pm 4\ 10^{-6}$ |
| 100 | 0.619 | $5.3\ 10^{-5} \pm 7.1\ 10^{-6}$ | $5.0\ 10^{-5} \pm 1.6\ 10^{-5}$ | $4.6\ 10^{-5} \pm 1.6\ 10^{-5}$ |
| $B_3$ [pkts] | $\rho_3$ | $IPLR_2$ | | |
| | | WFQ | SCFQ | DRR |
| 10 | 0.377 | $<10^{-6}$ | $2\ 10^{-5} \pm 8\ 10^{-6}$ | $3\ 10^{-5} \pm 6\ 10^{-6}$ |
| 20 | 0.490 | $<10^{-6}$ | $1.1\ 10^{-4} \pm 2.6\ 10^{-5}$ | $1.2\ 10^{-4} \pm 1.5\ 10^{-5}$ |
| 50 | 0.585 | $<10^{-6}$ | $2.5\ 10^{-4} \pm 3.5\ 10^{-5}$ | $2.3\ 10^{-4} \pm 1.1\ 10^{-5}$ |
| 100 | 0.619 | $<10^{-6}$ | $3.4\ 10^{-4} \pm 2.0\ 10^{-5}$ | $2.7\ 10^{-4} \pm 1.3\ 10^{-5}$ |

Tab. 5 Simulated $IPLR_2$ and $IPLR_3$ levels after loading CoS #3 to the values indicated in Tab. 2

| $IPLR_1$ | | $IPLR_2$ | | $IPLR_3$ | |
|---|---|---|---|---|---|
| PQ-SCFQ | PQ-DRR | PQ-SCFQ | PQ-DRR | PQ-SCFQ | PQ-DRR |
| $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $2.6\ 10^{-5} \pm 4.2\ 10^{-6}$ | $3.0\ 10^{-5} \pm 2.6\ 10^{-6}$ |
| $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $3.4\ 10^{-5} \pm 5.5\ 10^{-6}$ | $7.4\ 10^{-5} \pm 2.9\ 10^{-5}$ |
| $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $4.9\ 10^{-5} \pm 3.7\ 10^{-6}$ | $1.0\ 10^{-4} \pm 3.1\ 10^{-5}$ |
| $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $5.5\ 10^{-5} \pm 1.2\ 10^{-5}$ | $9.3\ 10^{-5} \pm 4.7\ 10^{-5}$ |

Tab. 6 Simulated $IPLR_1$, $IPLR_2$ and $IPLR_3$ levels after loading CoSs to the values indicated in Tab. 3; case for d=1.

| $IPLR_1$ | | $IPLR_2$ | | $IPLR_3$ | |
|---|---|---|---|---|---|
| PQ-SCFQ | PQ-DRR | PQ-SCFQ | PQ-DRR | PQ-SCFQ | PQ-DRR |
| $4.9\ 10^{-5} \pm 4\ 10^{-6}$ | $5.2\ 10^{-5} \pm 2\ 10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $1.54\ 10^{-4} \pm 2.5\ 10^{-5}$ | $2.3\ 10^{-4} \pm 6.1\ 10^{-5}$ |
| $4.8\ 10^{-5} \pm 10^{-6}$ | $5.3\ 10^{-5} \pm 2\ 10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $1.6\ 10^{-4} \pm 5.3\ 10^{-6}$ | $1.4\ 10^{-4} \pm 1.6\ 10^{-5}$ |
| $5.3\ 10^{-5} \pm 10^{-6}$ | $6.2\ 10^{-5} \pm 7\ 10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $1.36\ 10^{-4} \pm 1.2\ 10^{-5}$ | $1.12\ 10^{-4} \pm 8.8\ 10^{-5}$ |
| $6.2\ 10^{-5} \pm 4\ 10^{-6}$ | $6.4\ 10^{-5} \pm 2\ 10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $3.9\ 10^{-5} \pm 3.8\ 10^{-5}$ | $2.6\ 10^{-5} \pm 3.4\ 10^{-5}$ |

Tab. 7 Simulated $IPLR_1$, $IPLR_2$ and $IPLR_3$ levels after loading CoSs to the values indicated in Tab. 3; case for d=10.

We observe that different implementations of the same scheduler type (e.g. WFQ, SCFQ or DRR) give different values of IPLR. For example, for WFQ the value of $IPLR_2$ is lower than for SCFQ while the value of $IPLR_3$ is higher (see Tab. 5). This can be explained by differences in their realization resulting in different fairness in the small timescale. WFQ algorithm is more fair for CoS#2 and so $IPLR_2$ is decreased in the expense of CoS#3 for which $IPLR_3$ is increased. On the contrary, SCFQ algorithm is less fair for CoS#2 and so $IPLR_2$ is increased while the $IPLR_3$ is decreased as compared to the relevant values obtained with WFQ.

### 4. Conclusions

The performed simulation tests proved the correctness of the formulas in this sense that the target IPLR values were never violated when the system was loaded to the suggested limits. Since the formulas are correct and the calculated maximum admissible load is higher than in decomposition method, then it is beneficial to use the proposed approach.

## References

[1] Bąk A., Burakowski W., Ricciato F., Salsano S., Tarasiuk H. "A framework for providing differentiated QoS guarantees in IP-based network", Computer Communications 26 (2003) pp. 327-337.

[2] Tsolakou E., Nikolouzou E., Venieris I.S., "A study of QoS performance for peak time applications over Differentiated Services network", 7th IEEE Symposium on Computers and Communications, Taormina- Giardini Naxos, Italy, July 1-4, 2002.

[3] Zhang Z.-L., Towsley D. and Kurose J., "Statistical Analysis of Generalized Processor Sharing Scheduling Discipline," IEEE Journal of Selected Areas in Communications, 13(6): 1071-1080, August 1995.

[4] Lo Presti F., Zhang Z-L., Towsley D. "Bounds, approximations and applications for the two-queue GPS system", Infocom 1996.

[5] Mao S., Panwar S., Laptiotis G. "The Effective Bandwidth of Markov Modulated Fluid Process Sources with a Generalized Processor Sharing Server" , in Proceedings of IEEE GLOBECOM 2001, pp.2341-2346, San Antonio, TX, November 2001.

[6] Horvath G., Telek M., "Approximate analysis of two class WFQ systems", in Proc. of Workhop on Performability modeling of computer and communication systems – PMCCS 2003, pages 43-46, Arlington, IL, USA, September 2003.

[7] Walraevens J., Steyaert B., Bruneel H.. Performance analysis of the system contents in a discrete-time non-preemptive priority queue with general service times. Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL), 40(1-2), 2000.

[8] Babiarz J., Chan K., Baker F., "Configuration guidelines for DiffServ service classes", RFC 4594, August 2006.

[9] ITU-T Recommendation Y.1541, "Network performance objectives for IP-based services", ITU, May 2002.

[10] ITU-T Recommendation Y.1540, "Internet Protocol data communication service – IP packet transfer and availability performance parameters ", ITU, January 2002.

[11] Bruneel H., Kim B. G., "Discrete time models with application to ATM", Kluwer, 1993.

[12] Brandauer C., Burakowski W., Dąbrowski M., Koch B., Tarasiuk H., "AC algorithms in Aquila QoS IP network", European Transaction on Telecommunications, John Wiley & Sons, Inc., Vol. 16, No. 3, May-June 2005, pp. 225-232.

[13] Enriquez J. (eds.) et al., "EuQoS architecture update for phase 2", EuQoS project cosortium, Deliverable D122, February 2007, http://www.euqos.eu

[14] Szabo R. , Barta P. , Nemeth F. , Biro J. , Perntz C-G. "Call Admission Control in Generalized Processor Sharing (GPS) Schedulers Using Non-Rate Proportional Weighting of Sessions", IEEE INFOCOM 2000 - The Conference on Computer Communications, no. 1, March 2000, Israel, pp. 1243-1252.

[15] Guillemin F., Pinchon D., "Analysis of the Weighted Fair Queueing system with two classes of customers with exponential service times", Journal of Applied Probability, 2004.

[16] Borst S., Mandjes M., van Uitert M., "GPS queues with heterogeneous traffic classes", IEEE INFOCOM 2002.

[17] Ns-2 simulator http://isi.edu/~nsnam/ns

[18] Feller W. "Wstęp do rachunku prawdopodobieństwa", tom 1, PWN, 1987.

[19] Jin X., Min G., "Analytical Modelling of Hybrid PQ-GPS Scheduling Systems under Long-Range Dependent Traffic", Proc. IEEE 21st Int. Conference on Advanced Information Networking and Applications (AINA'2007), IEEE Computer Society Press, Niagara Falls, Canada, May 21-23, 2007.

[20] Cao J., Ramanan K., "A Poisson limit for buffer overflow probabilities", Proceedings of IEEE Infocom 2002, pp. 994-1003, New York.

[21] Gallager R.G., "Information theory and reliable communication", New York: Wiley, 1968.

[22] Gross D, Harris C. "Fundamentals of queueing theory", 3rd edition, 2002.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 213–224

# A Fluid-flow Approximation Model of TCP-NCR in wired-wireless networks *

TADEUSZ CZACHÓRSKI[a] KRZYSZTOF GROCHLA [a] FERHAN PEKERGIN [b]

[a]IITiS PAN, 44-100 Gliwice, ul. Bałtycka 5, Poland

[b]LIPN, Université Paris-Nord, 93 430 Villetaneuse, France

**Abstract:** The fluid-flow approximation models investigate with much success the dynamics and stability of TCP/RED connections. Their main assumption is that the fluctuations of variables characterizing the behaviour of the connections are relatively small, that enables the linearization of model and the use of traditional control analysis tools to obtain such measures as Bode gain, phase margins, tracking error or delay margin. In this article, preserving linear fluid-flow model, we propose its extension to the case when a network is composed of wired and wireless part. We consider a variant of TCP algorithm (TCP-DCR or its new version TCP-NCR) and fluid-flow differential equations representing the size of congestion window, mean queue at the bottleneck router and loss probability at a RED queue are supplemented with terms representing constant loss probability due to transmission in wireless part and probability that a fraction of these errors is recovered by a link level mechanism. The decrease of congestion window due to TCP mechanism is delayed to allow the link protocol to deal with the errors. The model considers the presence of uncontrollable UDP flows.

**keywords:** active queue management, RED, dynamics of TCP flows, unresponsible flows.

## 1. Introduction

A considerable work was done recently to capture the dynamics of TCP connections with the use of fluid flow approximation. The results include a model of a network with general topology [7], analysis of RED queue in the bottleneck router [4, 9], as well as its comparison with proportional-integral control [5]. In [6, 13] a mixture of TCP and UDP flows was introduced and in [11] the method is adapted to the analysis of satellite networks. In [12] an adaptation of this approach to several flavors of TCP is presented. Fluid flow approximation is relatively simple, compared to other analytical methods of queueing theory, and therefore well suited to analyze the behaviour of complex networks.

In general, the approach consists in solution of several equations. The first one concerns the dynamics of TCP congestion window $W$ and the second refers to the changes of the mean queue at the bottleneck router:

*This work was in part supported by Polish Ministry of Science and Higher Education grant 3T11C04530

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}p(t-R(t)) \quad (1)$$

$$\dot{q}(t) = \begin{cases} -C + \dfrac{K(t)}{R(t)}W(t), & q > 0 \\ \max\{0, -C + \dfrac{K(t)}{R(t)}W(t)\}, & q = 0, \end{cases} \quad (2)$$

where $\dot{x}$ denotes time-derivative and $W$ is the average TCP window size expressed in packets; $q$ is the average queue length in bottleneck router [packets]; $R(t)$ is round trip time [sec]; $C$ is link capacity [packets/sec]; $p$ is the probability of packet dropping or marking; $K$ is the number of TCP sessions active in the bottleneck router. The Eq. (1) may be extended to take into account the slow start procedure and time-out losses [9].

The set of equations modelling TCP flows dynamics is then linearized around the working point. The control loop may be then analyzed with standard tools appropriate to investigate the stability of a linear control system where the dynamics of its elements is represented by their transfer functions.

A way to incorporate in the model the unresponsive flows is to modify Eq. (2) in the following way [6]. Let $l(t)$ denotes the traffic of responsive flows and $u(t)$ of the unresponsive ones,

$$\dot{q}(t) = \begin{cases} -C + \dfrac{K(t)}{R(t)}W(t) + u(t), & q > 0 \\ \max\{[0, -C + \dfrac{K(t)}{R(t)}W(t) + u(t)\}, & q = 0. \end{cases} \quad (3)$$

It is supposed in [6] that $u(t)$ is a stationary process with mean $u_0$ and its variations are on a finer time-scale, hence it may be regarded as constant and it results in diminishing the bandwidth seen by responsive flows:

$$\dot{q}(t) = \begin{cases} -(C - u_0) + \dfrac{K(t)}{R(t)}W(t), & q > 0 \\ \max\{0, -(C - u_0) + \dfrac{K(t)}{R(t)}W(t)\}, & q = 0. \end{cases} \quad (4)$$

The linearization around working point $(W_{l0}, p_0, q_0, u_0)$ gives transfer functions, expressed here in terms of their Laplace transforms

$$W(s) = -P_{win}(s)e^{-sR_0}p(s), \qquad l(s) = \frac{K}{R_0}W(s), \quad (5)$$

$$q(s) = P_{que}(s)\left[l(s) + u(s)\right], \qquad p(s) = q(s)C_{aqm}(s) \quad (6)$$

where

$$P_{win}(s) = \frac{\frac{R_0 C_{eff}^2}{2K^2}}{s + \frac{2K}{R_0^2 C_{eff}}}, \qquad P_{que}(s) = \frac{1}{s + \frac{C_{eff}}{C}\frac{1}{R_0}}, \qquad C_{eff} = C - u_0.$$

and $C_{aqm}(s)$ is the Laplace transform of transfer function of the AQM controller. The above equations are matched with others which define the probability of dropping or marking a packet as a function of the mean queue size. In the case of classical RED algorithm, the dropping probability is defined by a curve having three parameters: $min_{th}$, $max_{th}$ and $max_p$,

$$p_d = \begin{cases} 0 \quad \text{if} \quad avg \ \leq \ min_{th} \\ \dfrac{avg - min_{th}}{max_{th} - min_{th}} * max_p \quad \text{if} \quad min_{th} < avg \leq max_{th} \\ 1 \quad \text{if } max_{th} < avg, \end{cases} \tag{7}$$

where the moving average *avg* is calculated at the arrival of each packet and represents a low-pass filter for the actual queue length

$$avg = \begin{cases} (1-w) * avg \ + \ w * n \quad \text{if queue is not empty} \\ (1-w)^{\frac{\mu}{\lambda}} * avg \quad \text{if queue is empty,} \end{cases}$$

where $w$ is a fixed (small) parameter and $n$ is the instantaneous queue size.

The process of taking the moving average (denoted below by $x$) may be modelled as in [9]

$$\frac{dx}{dt} = \log_e(1-w)/\Delta - \log_e(1-w)/\Delta q(t) \tag{8}$$

where $\Delta$ is the interarrival time of packets and is taken as $\Delta = 1/C$.

Hence, the transfer function of RED mechanism having changes of current queue $\delta q$ at the entrance and changes of packet loss $\delta p$ as the output has the form

$$C_{aqm}(s) = L_{red}\frac{k}{k+s}$$

where

$$k = -\ln(1-w)/\Delta = -C\ln(1-w) \qquad \text{and} \qquad L_{red} = \frac{max_p}{max_{th} - min_{th}}.$$

The above equation may be replaced by another if the AQM algorithm changes.

This way a control loop schema as in Fig. 2 is formulated and analysed. Below, we present some notions on the stability analysis following [10]. The stability of the system may be investigated with the use of Nyquist criterion. In general, the closer the transfer function $G(j\omega)$ representing the whole control loop transfer functions, comes to encircling

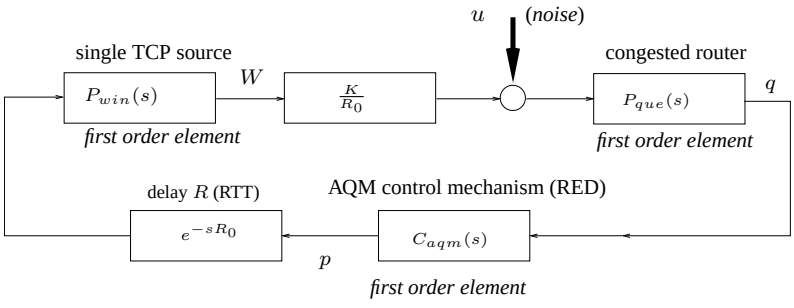



Fig. 1. Control schema 1

the $(-1 + j0)$ point, the more oscillatory is the system response. The closeness of $G(j\omega)$ locus to the $(-1 + j0)$ point can be used as a measure of the margin of stability. It is common practice to represent the closeness in terms of *phase margin* and *gain margin*. *Phase margin* is the amount of additional phase lag at the gain crossover frequency required to bring the system to the verge of instability. The gain crossover frequency is the frequency at which $|G(j\omega)|$, the magnitude of open-loop transfer function, is unity. The phase margin $\gamma$ is $180\deg$ plus the phase angle $\varphi$ of the open-loop transfer function at the gain crossover frequency, or $\gamma = 180\deg + \varphi$. On the Nyquist diagram, a line may be drawn from the origin to the point at which the unit circle crosses the $|G(j\omega)|$ locus. The angle from the negative real axis to this line is the phase margin. The phase margin is positive for $\gamma > 0$. For a minimum phase system to be stable, the phase margin must be positive [10].

Figures 3 - 5 illustrate the influence of the transmission and control parameters on the shape of Nyquist plots. In plots, similarly as in [9] the presence of transfer function $\Delta(s)$, see fig. 2 was neglected and considered as a source of noise

In this numerical example $C = 12500$ packets/s, $K = 60$, and RED parameters are $p_{max} = 0.01$, $max_{th} = 200$, $min_{th} = 100$, $w = 0.0001$. Delay $R$ is varied between 10 ms and 100ms, $C_{eff}$ is equal $0.25C$, $0.5C$, and $0.75C$.

In section 2. we propose an extension of this model to capture the performance of TCP algorithm at wired-wireless environment.

## 2. TCP in wireless environment, TCP-DCR

In traditional TCP implementation, like New Reno, even if the wireless channel recovery mechanism is able to retransmit the packet, it is considered as a packet loss by the transport layer, due to the high delay. In wireless networks the packet loss often occur due to transmission errors, in typical cases with a ratio going up to a few percent. The classical TCP implementation does not work well in this case. That is why some modification of the classi-

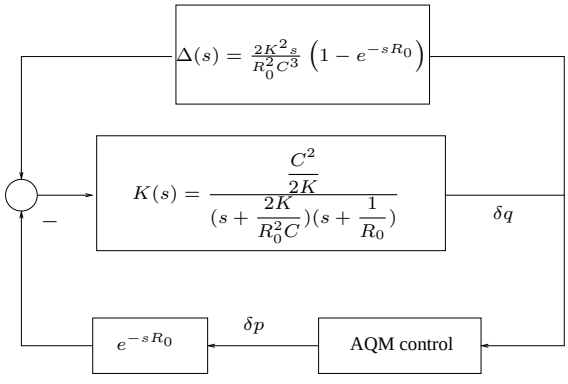



Fig. 2. Control schema of TCP Reno connections, see [9]

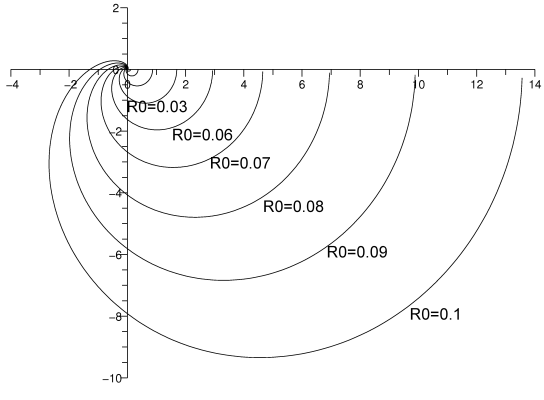



Fig. 3. Reno connections, $G(jm)$ for various values of delay $R$, stable and unstable cases

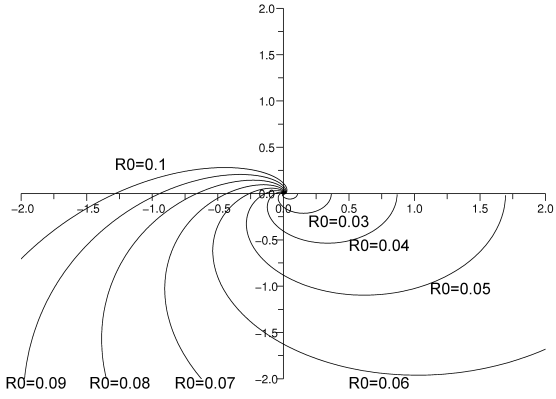



Fig. 4. Nyquist plot for various delays $R_0$ (fragment)

cal New Reno algorithm were proposed, e.g. TCP Westwood [8]. Recently, several solutions have been proposed to improve the performance of TCP over wireless networks. In general, these solutions are classified in four categories *split connection* approaches, *link layer scheme, explicit loss notification* approaches and *receiver-based* approaches.

Here, we concentrate on TCP-Delayed Control Rate (or Delayed Congestion Response) TCP-DCR, [1]. It is a modification of the TCP protocol created to improve its robustness to channel errors in wireless network. The TCP-DCR is based on the idea of allowing the link level mechanism to recover the packets lost due to channel errors. This is done by delaying the triggering of congestion response (fast retransmission-recovery) algorithms due to reception of dupacks or due to time-out of the retransmission timer, for a small bounded period of time $\tau$ to allow the link level retransmission to recover the loss due to channel errors. The choice of $\tau$ is critical, makes a trade off between unnecessarily inferring congestion, and unnecessarily waiting for a long time before retransmitting a lost packet. Analytical considerations in [1] indicate that both upper and lower limit for $\tau$ is RTT, hence the best choice is $\tau = RTT$.

These principles then evolved in document [2] where the non-congestion robustness (NCR) is discussed and two variants of TCP-NCR are proposed: (i) *Careful Limited Transmit* which calls for reducing the sending rate at approximately the same time implementations reduce the congestion window, as defined by RFC 2581, while at the same time withholding a retransmission (and the final congestion determination) for approximately one RTT. (ii) *Aggressive Limited Transmit* that calls for maintaining the sending rate in the face of duplicate ACKs until TCP concludes that a segment is lost and needs to be retransmitted (which TCP-NCR delays by one RTT when compared with current loss recovery schemes).

According to TCP-DCR philosophy, the eqs. (1,2) are modified:

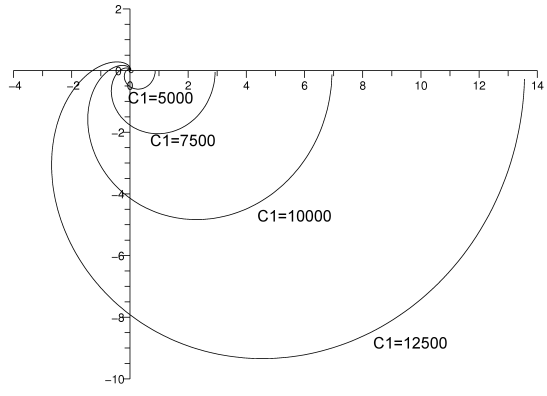



Fig. 5. Reno connections, $G(jm)$ for various values of effective band $C_{eff}$, stable and unstable cases

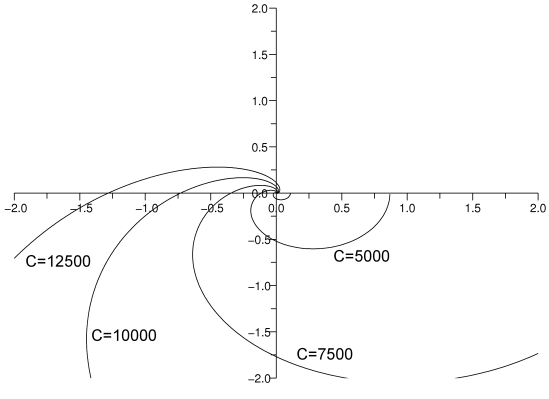



Fig. 6. Reno connections, $G(jm)$ for various values of effective band $C_{eff}$, stable and unstable cases (fragment)

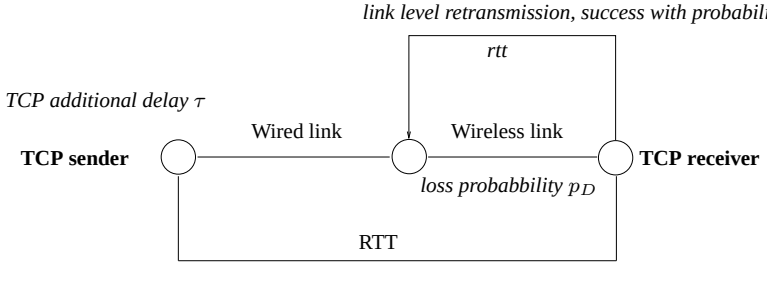



Fig. 7. DCR parameters

$$\begin{aligned}\dot{W}(t) &= \frac{1-P_D}{R(t)} + \frac{P_D\alpha}{R(t)+rtt} - \frac{W(t)}{2} \times \\ &\quad \times \frac{W(t-R(t)-\tau)}{R(t-R(t)-\tau)}[p(t-R(t)-\tau) + P_D(1-\alpha)], \end{aligned} \tag{9}$$

$$\dot{q}(t) = \begin{cases} -C + \dfrac{K(t)}{R(t)(1-P_D) + (R(t)+rtt)P_D\alpha}W(t), & q > 0 \\ \max\{0, -C + \dfrac{K(t)}{R(t)(1-P_D) + (R(t)+rtt)P_D\alpha}W(t)\}, & q = 0, \end{cases} \tag{10}$$

where $P_D$ is congestion-independent loss probability in wireless part of the network, *rtt* is time after which the wireless protocol is able to recover from an error with probability $\alpha$.

We linearized eqs. (9) and (2) in the same way as eqs. (1) and (2) were linearized in [4]. First we assume that the number of TCP sessions and link capacity are constant, i.e., $K(t) \equiv K$ and $C(t) \equiv C$. Taking $(W, q)$ as the state and $p$ as input, the operating point $(W_0, q_0, p_0)$ is then defined by $\dot{W} = 0$ and $\dot{q} = 0$.

To proceed with linearization of (1), we ignore the dependence of the time-delay argument $t - R$ on queue-length $q$, and assume it fixed to $t - R_0$. On the other hand, we retain the dependence of round-trip time on queue length in the dynamic's parameters. We have two functions:

$$\begin{aligned}\dot{W}(t) &= f_1\Big(W(t), W(t-R(t)-\tau), q(t), q(t-R(t)-\tau), p(t-R(t)-\tau)\Big) \\ &= f_1(W, W_{R+\tau}, q, q_{R+\tau}, p_{R+\tau}) \\ &= \frac{1-P_D}{q/C+T_p} + \frac{P_D\alpha}{q/C+T_p+rtt} - \frac{W}{2}\frac{W_{R+\tau}}{q_{R+\tau}/C+T_p}[p_{R+\tau} + P_D(1-\alpha)] \\ \dot{q}(t) &= f_2\Big(q(t), W(t)\Big) = f_2(q, W)\end{aligned}$$

Linearization of a function $f(x_1, x_2, \ldots, x_n)$ is done around a working point $(x_{1,0}, x_{2,0}, \ldots, x_{n,0})$ for small changes of arguments

$$\delta f = \sum_{i=1}^{n} \frac{\partial f(x_1, x_2, \ldots, x_n)}{\partial x_i}\Big|_{(x_1=x_{1,0}, x_2=x_{2,0}, \ldots, x_n=x_{n,0})} \delta x_i$$

The working point is defined by the condition $f_1(.) = f_2(.) = 0$, hence by relations

$$\frac{1-P_D}{R_0} + \frac{P_D\alpha}{R_0+rtt} - \frac{W_0^2}{2R_0}[p_0 + P_D(1-\alpha)] = 0$$

and

$$-C + \frac{KW_0}{R_0(1-P_D) + (R_0+rtt)P_D\alpha} = 0$$

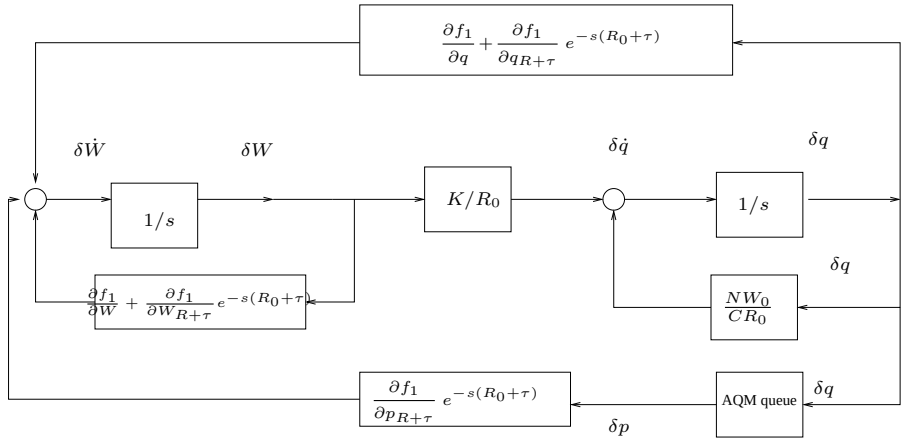



Fig. 8. Block diagram for linearized TCP-DCR control

We obtain

$$\begin{aligned} \delta\dot{W} &= \frac{\partial f_1}{\partial W}\delta W + \frac{\partial f_1}{\partial W_{R+\tau}}\delta W_{R+\tau} + \frac{\partial f_1}{\partial p_{R+\tau}}\delta p_{R+\tau} + \frac{\partial f_1}{\partial q}\delta q + \frac{\partial f_1}{\partial q_{R+\tau}}\delta q_{R+\tau} \\ \delta\dot{q} &= \frac{\partial f_2}{\partial W}\delta W + \frac{\partial f_2}{\partial q}\delta q \end{aligned}$$

Fig. 8 displays the block diagram of the control loop resulting from this linearization.

In numerical example below we assumed the following parameters: link capacity $C = 12500$ packets/s, number of connections $K = 60$; RED parameters $p_{max} = 0.01$, $max_{th} = 200$, $min_{th} = 100$ $w = 0.0001$ at the working point loss probability due to congestion is $p_0 = 0.05$, the size of congestion widow $W_0 = 10$, $R_0 = 0.05$; other parameters such as loss probability for wireless part $P_D$, , $\tau$, $rtt$, probability of successful retransmission in wireless part $\alpha$, are varied as indicated in figures. Figures 9 - 11 represent some exemplary curves $G(j\omega)$ as a function of mentioned above parameters. Fig. 12 compares plots for two versions od NCR protocol.

## 3. Conclusions

The presented model adapts the fluid-flow model of TCP/RED congestion control mechanism to the case when a network is composed of wired and wireless part. A DCR-TCP and its successing two variants of NCR-TCP protocol are considered. Presented numerical examples illustrate the influence of several parameters of these protocols on the stability of connections. Further systematic numerical investigations to determine stability areas are needed.

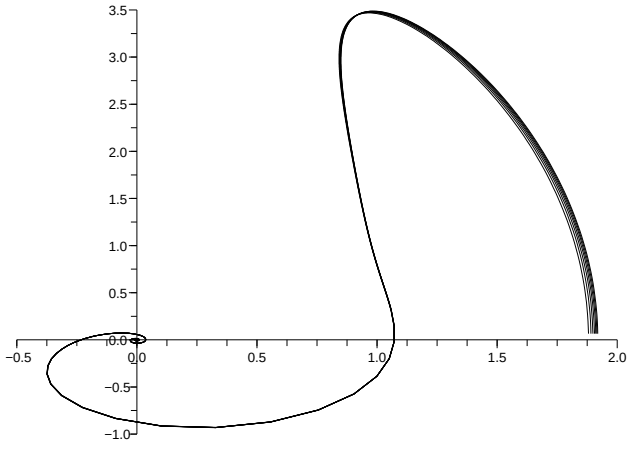



Fig. 9. $G(jm)$, the influence of the changes of $rtt = 0.1RTT, \ldots, 0.9RTT$

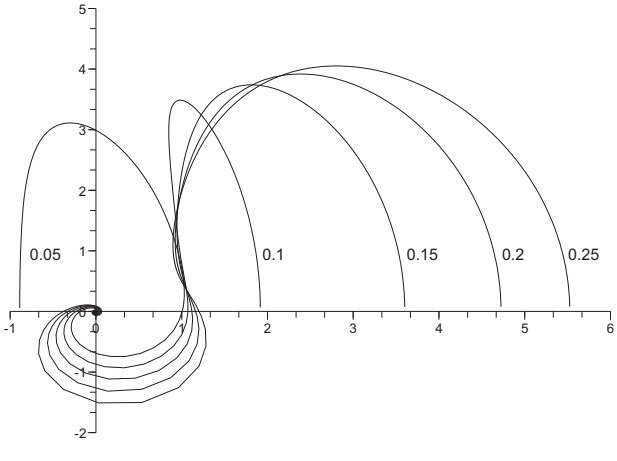



Fig. 10. $G(jm)$, the influence of losses in wireless part $P_D = 0.05, 0.10, \ldots 0.25$

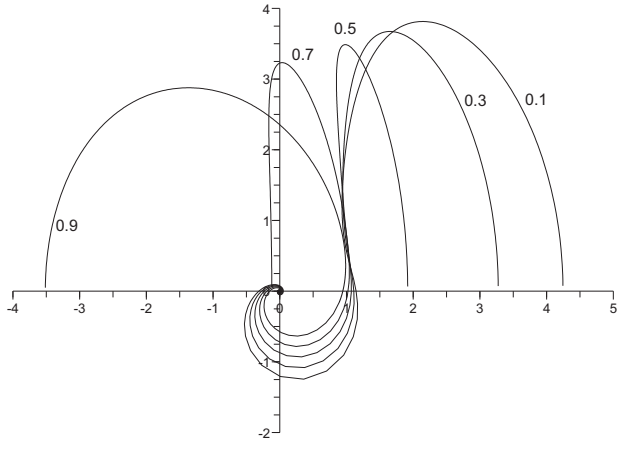



Fig. 11. $G(jm)$, the influence of the probability $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ of successful retransmission in the wireless part

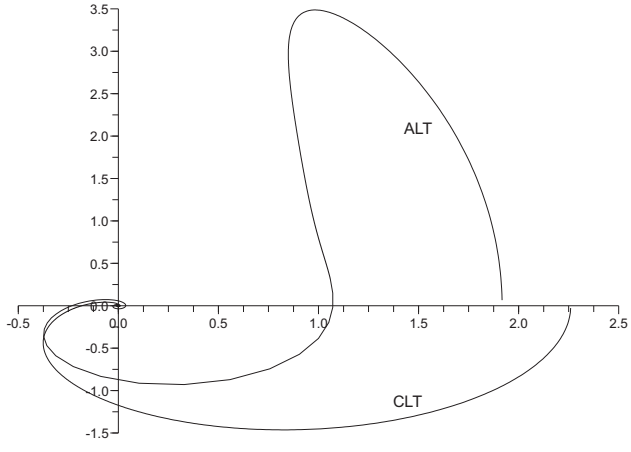



Fig. 12: $G(jm)$ The influence of multiplicative factor $a$ to decrease the congestion window in case of a transmission ($a = 1/2$ in case of *aggressive limited transmit NCR* (ALT), and $a = 2/3$ in case of *careful limited transmit NCR* (CLT)

## References


[1] S. Bhandarkar, N. Sadry, A.L.N. Reddy, N. Vaidya, *TCP-DCR: A novel protocol for tolerating wireless channel errors*, Technical Report TAMU-ECE-2003-01, February 2003.

[2] S. Bhandarkar, A.L.N. Reddy, M. Allman, E. Blanton, *Improving the Robustness of TCP to Non-Congestion Events*, Network Working Group Request for Comments: RFC 4653

[3] J. Chen, F. Paganini, R. Wang, M. Y. Sanadidi, M. Gerla, *Fluid-flow analysis of TCP Westwood with RED*, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 50, Iss. 9, June 2006.

[4] C. V. Hollot, Vishal Misra, Don Towsley et al. *A control theoretic analysis of RED*, Proc of IEEE/INFOCOM, 2001.

[5] C. V. Hollot, V. Misra, D. Towsley, W.B. Dong, *Analysis and Design of Controllers for AQM Routers Supporting TCP Flows*, IEEE Transactions on Automatic Control, special issue on Systems and Control Methods for Communication Networks, vol. 47, no. 6, 2002.

[6] C. V. Hollot, Y. Liu, V. Misra, D. Towsley et al. *Unresponsive flows and AQM Performance*, Proc. of IEEE INFOCOM 2003.

[7] Y. Liu, F. Lo Presti, V. Misra, Y. Gu, *Fluid Models and Solutions for Large-Scale IP Networks*, ACM/SigMetrics 2003.

[8] S. Mascolo, C. Casetti, M. Gerla, M.Y. Sanadidi, R. Wang: *TCP Westwood Bandwidth Estimation for Enhanced Transport over Wireless Links*, in: Mobile Computing and Networking, pp.287-297, 2001.

[9] V. Misra, W.-B. Gong, D. Towsley: *Fluidbased Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED*, ACM SIGCOMM 2000.

[10] K. Ogata, *Modern Control Engineering*, Prentice Hall of India, New Dehli 1977.

[11] M. Sridharan, et al, *Tuning RED parameters in Satellite Networks Using Control Theory*, Proc. of Performance and Control of Next Generation Communication Networks, SPIE Vol. 5244, Orlando Florida, September 7-11, pp.145-153.

[12] R. Srikant, *The Mathematics of Internet Congestion Control*, Springer Series: Systems and Control: Foundations and Applications, Berlin 2004.

[13] Wang Li, LI Zeng-zi, Chen Yan-ping, Xue Ke, *Fluid-Based stability Analysis of Mixed TCP and UDP Traffic under RED*, Proc of the 10th Int. Conf on Engineering of Complex Computer Systems, (ICECCS), 2005.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 225–234

# On the Multistage Packet Processing

PAWEŁ ŚWIĄTEK [a]

[a]Institute of Information Science and Engineering
Wrocław University of Technology
pawel.swiatek@pwr.wroc.pl

**Abstract:** In this paper a model of multistage packet processing system is defined in terms of the control theory. Basing on the proposed model, problems of single and multistage packet scheduling are formulated. Moreover, the problem of coordination of local control (scheduling) algorithms in the multistage scheduling system is defined. Since formulated problems are NP-hard, optimal algorithms cannot be applied to real-time traffic control. Instead, one can use on-line approaches, which approximate the optimal solution. Therefore, we show how to adapt existing single stage scheduling algorithms to the *on-line* version of multistage packet scheduling problem. Finally, we give some remarks concerning the application of artificial intelligence methods to the considered problem. The discussion is followed by illustrative simulation example, which confirms the efficiency of proposed solutions.

**Keywords:** Packet scheduling, traffic flow control, Quality of Service

## 1. Introduction

Providing quality of service (QoS) in computer communication packet-switched networks is strongly related to the problem of traffic flow control in the network node. Efficient traffic flow control requires that the aggregated stream of packets incoming into the network node is decomposed into separate traffic classes, which require different types of service [6]. Depending on the traffic class there are different traffic parameters, that are taken into account during the evaluation of the quality of service e.g.: jitter for voice transmission, delay for real-time systems and packet loss ratio for data transfer. Additional issues of QoS management include: traffic shaping, link sharing and fairness.

In this paper a new model of multistage packet processing system is proposed. Processing of the incoming stream of packets consists of three stages (fig. 1). First, aggregated stream of packets is divided into substreams (e.g. connections) and each substream is assigned for further processing to one of $M$ parallel processors $O_m, m \in \{1, ..., M\}$ on second stage. Finally, output of the second stage processing is aggregated again into single stream and forwarded to the network by device $O_{M+1}$.

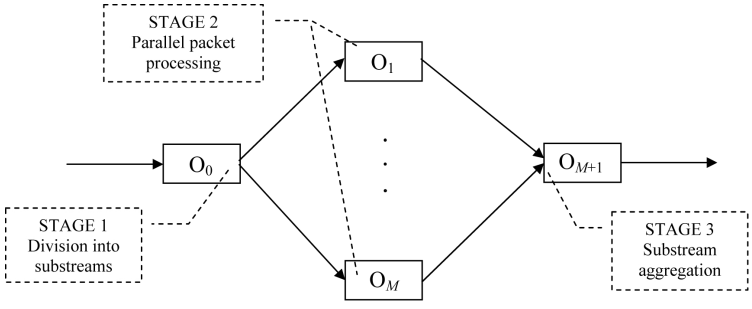



Fig. 1. Multistage packet scheduling system.

Even though there exist a number of models and algorithms for single stage packet processing, the proposed approach has some advantages. First of all, parallel processing significantly increases systems throughput and reliability. Moreover, in such a system there is a possibility to make use of specialized devices adapted to specific features of certain packet substreams (traffic classes), and thus able to perform certain task better (faster, more accurate) than universal devices.

In this paper, basing on the control theory, problems of single- and multistage packet scheduling in the network node are defined. Since formulated problems are NP-hard, optimal algorithms cannot be applied to real-time traffic control. Instead, one can use *on-line* approaches, which approximate the optimal solution. Therefore, we show how to adapt existing single stage scheduling algorithms to the on-line version of multistage packet scheduling problem. Finally, we give some remarks concerning the application of artificial intelligence methods to the considered problem.

## 2. Single stage packet scheduling

Generally, traffic flow control in the network node consist in scheduling packets from different traffic classes. Let $K$ be the number of traffic classes. Packets belonging to each traffic class $k$, $k \in \{1, \ldots, K\}$ flow into the node according to a certain probability distribution with the mean intensity $\lambda^{(k)}$ and wait for service in $k$-th queue (fig. 2).

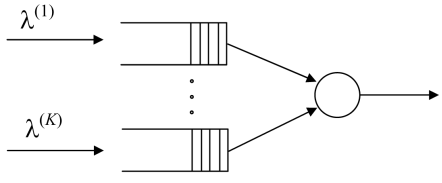



Fig. 2. Single stage packet scheduling.

Moreover, each class $k$ is characterized by priority $p^{(k)}$ and quality of service criterion

$q^{(k)}$. Let $\mathbf{p} = [p^{(1)}, \ldots, p^{(K)}]$ and $\mathbf{q} = [q^{(1)}, \ldots, q^{(K)}]$ are vectors of priorities and QoS criteria respectively.

The task of decision algorithm is to schedule packets from different classes (queues) such that quality of service guaranties are satisfied, i.e. certain criterion $Q(\mathbf{q}, \mathbf{p})$ is minimized.

Foregoing packet scheduling system can be modelled as the input-output control plant ***CP*** [3] (fig. 3).

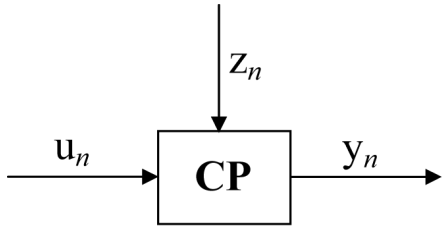



Fig. 3. Model of the single stage packet scheduling system.

Let $\mathbf{u}_n = [u_n^{(1)} \ldots u_n^{(K)}]^T$ (where $n$ is the number of control step) be the input (decision) vector of the control plant, where $u_n^{(k)} = 1$ if the packet, which is going to be serviced belongs to $k$-th class and $u_n^{(i)} = 0$ for $i \in \{1, \ldots, K\} \backslash \{k\}$. Moreover, let intensities of each traffic class are the disturbance $\mathbf{z}_n = [\lambda_n^{(1)} \ldots \lambda_n^{(K)}]^T$. Finally, let the vector of temporary values of the quality of service criteria of each class is the output of the system $\mathbf{y}_n = [q_n^{(1)} \ldots q_n^{(K)}]^T$. Since the state $\mathbf{x}_n = [x_n^{(1)} \ldots x_n^{(K)}]^T$ of the control plant ***CP*** is precisely defined by lengths of queues $k$, then control plant ***CP*** can be described by the following state equation

$$\begin{cases} \mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{u}_n + \mathbf{z}_n \\ \mathbf{y}_n = F(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n) \end{cases} \tag{1}$$

where function $F$ is (in general) unknown, however values of $\mathbf{y}_n$ can be measured. The quality of control (service) $Q_n$ is evaluated according to certain criterion function $\varphi$, and $Q_n = \varphi(\mathbf{y}_n, \mathbf{p})$.

For such a system we formulate the following control (decision making) problem.

**Problem 1 (P1):**

**Given:** $\varphi$, $F$, $\mathbf{p}$, $\mathbf{x}_0$, $N$

**Find:** The sequence of optimal decisions $(\mathbf{u}_0^*, \ldots, \mathbf{u}_{N-1}^*)$ such that quality criterion function $\sum_{n=1}^{N} Q_n$ is minimized:

$$(\mathbf{u}_0^*, \ldots, \mathbf{u}_{N-1}^*) = \arg \min_{\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}} \sum_{n=1}^{N} Q_n \tag{2}$$

Since

$$Q_n = \varphi(\mathbf{y}_n, \mathbf{p}) = \varphi(F(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n), \mathbf{p}) \hat{=} \Phi(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n, \mathbf{p}) \tag{3}$$

then (2) reduces to

$$(\mathbf{u}_0^*, \ldots, \mathbf{u}_{N-1}^*) = \arg \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}} \sum_{n=1}^{N} \Phi(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n, \mathbf{p}) \tag{4}$$

There is possibility to find optimal solution of some special cases of problem P1 by applying dynamic programming procedure, in general however, this problem belongs to the class of NP-hard problems and it is highly unlikely to solve it in acceptable time. The problem becomes more complex as we note, that function $F$ (and in consequence function $\Phi$) is unknown and disturbances $\mathbf{z}_n$ (which are the intensities of traffic classes) are random. Therefore, in practice, only heuristic on-line algorithms are applied, which approximate optimal solution by minimizing the temporary quality of service criterion $Q_n$.

Nowadays, there exist a number of efficient single stage packet scheduling algorithms. The simplest are based on the *Weighted Round Robin*, e.g. WRR with adaptively changing weights based on the reinforcement learning [10]. This approach belongs to the wider group of algorithms - *Class Based Queuing* [5]. Next, in [11] author presents commonly used link sharing algorithm *Token Bucket* and its hierarchical version *Hierarchical Token Bucket*. Another approach to traffic flow control are methods based on *fairness* principle (e.g.: Weighted Fair Queuing [8], Worst-case Fair Weighted Fair Queuing [1], Hierarchical Packet Fair Queuing [2]). Finally, a significant improvement in the quality of packet scheduling was achieved after introducing new scheduling approach based on *service curves* (i.e.: *Fair Service Curve* [7] and *Hierarchical Fair Service Curve* [9]).

## 3. Multistage packet scheduling

Consider the multistage packet scheduling system from figure 1, where each object $O_m$, $m \in \{1, \ldots, M+1\}$ has the structure as shown on figure 2.

Traffic incoming into the network node is divided into substreams (e.g. connections) and directed for further processing by dispatcher $O_0$. Next, parallel substreams are processed separately on devices $O_m$, $m \in \{1, \ldots, M\}$ and finally merged into one output stream on processor $O_{M+1}$.

The task of such a system is to schedule packets in such a way, that certain global quality of service criterion is minimized.

On figure 4 the multistage packet scheduling system as the input-output control system is presented.

Let $\mathbf{X}_n = [\mathbf{x}_{1,n} \ldots \mathbf{x}_{M+1,n}]^T$ and $\mathbf{U}_n = [\mathbf{u}_{1,n} \ldots \mathbf{u}_{M+1,n}]^T$ are respectively state and decision vectors of devices $O_m$, $m \in \{1, \ldots, M+1\}$ and $\mathbf{Z}_n = [\mathbf{z}_{01,n} \ldots \mathbf{z}_{0M,n}\ \mathbf{z}_{1M+1,n} \ldots \mathbf{z}_{MM+1,n}]^T$ is the vector of disturbances. Outputs $\mathbf{y}_{m,n}$ are vectors of temporary values of local quality of service of each traffic class, and $\mathbf{y}_n$ is the vector of temporary values of quality of service of each class on the output of whole system. The global quality of service $Q_n$ is calculated according to certain criterion function $Q_n = \varphi(\mathbf{y}_n, \mathbf{p})$.

The problem of packet scheduling in such a system can be formulated as follows.

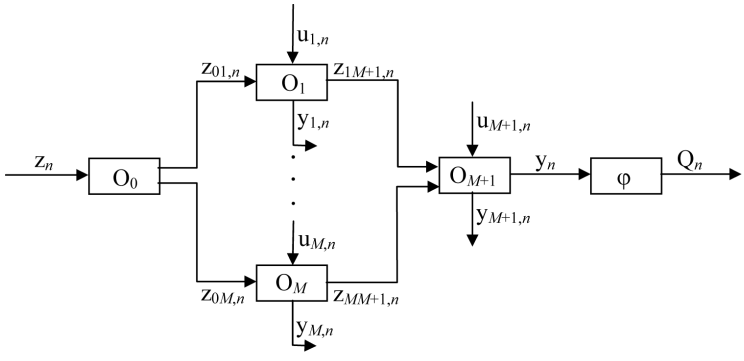



Fig. 4. Model of the multistage packet scheduling system.

**Problem 2 (P2):**
**Given:** $\varphi$, $F$, $\mathbf{p}$, $\mathbf{X}_0$, $N$
**Find:** The sequence of optimal decisions $(\mathbf{U}_0^*, \ldots, \mathbf{U}_{N-1}^*)$ such that quality criterion function $\sum_{n=1}^{N} Q_n$ is minimized:

$$(\mathbf{U}_0^*, \ldots, \mathbf{U}_{N-1}^*) = \arg \min_{\mathbf{U}_0, \ldots, \mathbf{U}_{N-1}} \sum_{n=1}^{N} Q_n \tag{5}$$

what after transformation analogical to (3) gives

$$(\mathbf{U}_0^*, \ldots, \mathbf{U}_{N-1}^*) = \arg \min_{\mathbf{U}_0, \ldots, \mathbf{U}_{N-1}} \sum_{n=1}^{N} \Phi(\mathbf{X}_n, \mathbf{U}_n, \mathbf{Z}_n, \mathbf{p}) \tag{6}$$

Note, that the difference between single- and multistage packet scheduling is that decisions made on the second stage (parallel processing) influence the quality of service on the third stage. In fact, single stage system from figure 2 is a part of the multistage system from figure 1 and in consequence problem P1 is a subproblem of problem P2.

Obviously, for the same reasons as for the single stage case, determination of solution (6) is not possible and one should apply on-line algorithms, which minimize the temporary quality of service criterion $Q_n$ in successive control steps.

## 4. Improving the performance of the system

Since each of devices $O_m$, $m \in \{1, \ldots, M+1\}$ can be treated as the single stage scheduling system, it is possible to apply known packet scheduling strategies as the local control algorithms in the multistage system. Note, however, that such an approach can yield only locally optimal solutions, which in general are worse from global point of view. Only for simple cases and special forms of criterion function it can be shown, that locally and

globally optimal solutions are the same. On the other hand, calculating global solution may be too complex (time-consuming) to be applied in real-time control systems.

In order to improve performance of the system an upper level control algorithm may be used, which would coordinate [4] local scheduling algorithms $H_m$. The task of such a coordinator would be to calculate new parameters of decision algorithms basing on measured values of systems inputs, outputs and states. In such a case coordinator acts as an adaptator. On figure 5 the proposed multistage packet processing system, which includes adaptation (block $A$) is presented. Due to clarity reasons a substitute plants $OZ_m$ (depicted on figure 6) were introduced.

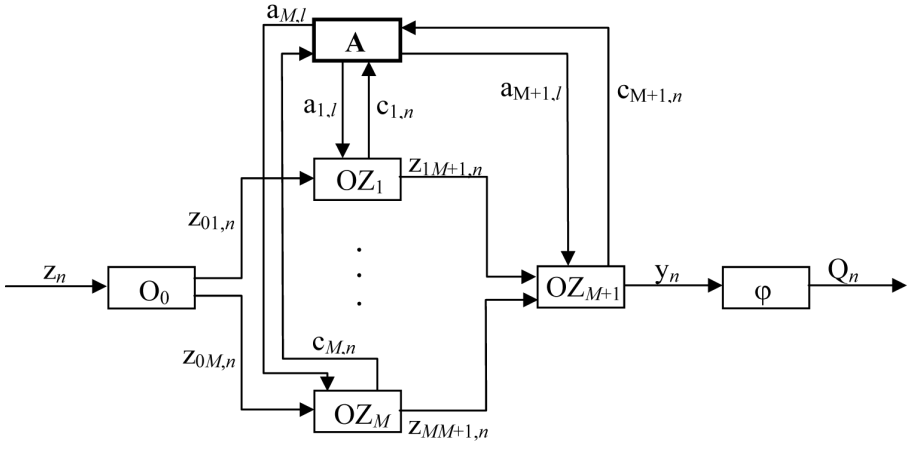



Fig. 5. Multistage packet scheduling with adaptation.

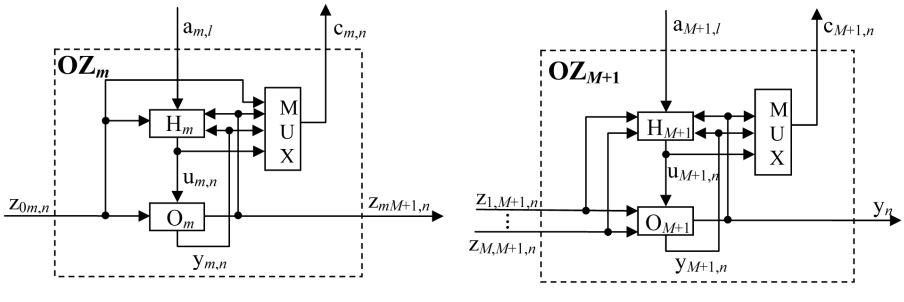



Fig. 6. Structure of substitute plants $OZ_m$.

Let $l$ be the adaptation step number which lasts $N$ basic control steps. Moreover, let $\mathbf{C}_l = [\mathbf{c}_{l\cdot N-N+1} \dots \mathbf{c}_{l\cdot N}]$ be the matrix of systems parameters during the $l$-th adaptation step, where vectors $\mathbf{c}_n = [\mathbf{c}_{1,n} \dots \mathbf{c}_{M+1,n}]^T$ describe system in $n$-th moment and $n = l \cdot N - N + 1, \dots, l \cdot N$. Additionally, denote by $\mathbf{a}_l = [\mathbf{a}_{1,l} \dots \mathbf{a}_{M+1,l}]^T$ the vector of new parameters of local control (scheduling) algorithms $H_m$, $m = 1, \dots, M+1$. The task of adaptation consists

in that for the sequence of measured systems parameters $\mathbf{C}_l$ one should find such a new vector of control algorithm parameters $\mathbf{a}_l$ that minimizes the value of the global quality of service criterion $\sum_{n=l\cdot N-N+1}^{l} Q_n = \sum_{n=l\cdot N-N+1}^{l} \Theta(\mathbf{c}_n, \mathbf{a}_l) \hat{=} \bar{\Theta}_N(\mathbf{C}_l, \mathbf{a}_l)$ during $N$ control steps.

For such described system the following problem of adaptation (coordination) can be formulated.

**Problem 3 (P3):**

**Given:** $\Theta$, $N$, $\mathbf{C}_l$

**Find:** The vector of optimal parameters $\mathbf{a}_l^*$ such that quality criterion function $\sum_{n=l\cdot N-N+1}^{l} Q_n = \sum_{n=l\cdot N-N+1}^{l} \Theta(\mathbf{c}_n, \mathbf{a}_l) \hat{=} \bar{\Theta}_N(\mathbf{C}_l, \mathbf{a}_l)$ is minimized:

$$\mathbf{a}_l^* = \arg\min_{\mathbf{a}_l} \bar{\Theta}_N(\mathbf{C}_l, \mathbf{a}_l) \tag{7}$$

Unfortunately, function $\Theta$, and what follows, function $\bar{\Theta}$ are not known. It is merely possible to measure consecutive values of criterion $Q_n$ for varying parameters of the system. Thus, analytical solution of the adaptation problem **P3** cannot be found.

There are, however, another two approaches to handle that problem. In the first method we assume certain function $Q_n = \Theta(\mathbf{c}_n, \mathbf{a}_l; \mathbf{b})$ and identify its parameters $\mathbf{b}$ during run of the system, what allows us to solve problem (7) by applying the successive approximations method:

$$\mathbf{a}_{l+1} = \mathbf{a}_l - \mathbf{K} \cdot \mathbf{w}_l \tag{8}$$

where

$$\mathbf{w}_l = \nabla_{\mathbf{a}} \bar{\Theta}_N(\mathbf{C}_l, \mathbf{a}; \mathbf{b}_l)|_{\mathbf{a}=\mathbf{a}_l} \tag{9}$$

and $\mathbf{b}_l$ is a vector of identified parameters of the assumed model $\Theta$.

Such an approach, which treats whole system as the "black box" may sometimes yield questionable results, because assumed model not always allows us to take into consideration dynamics of the system. Therefore, the choice of the model $\Theta$ has a major impact on systems performance.

Another approach, which is more accurate (and of course computationally more complex) takes advantage of the fact, that even though the function $\Theta$ is not known, we have the description of the system in the form of applied algorithms. Thus, information about the temporary state of the system $\mathbf{c}_n$ allows us to calculate the value of the criterion function $Q_n = \Theta(\mathbf{c}_n, \mathbf{a})$ for arbitrary values of parameters $\mathbf{a}$ by means of computer simulation $S$.

$$Q_n = S(\mathbf{c}_n, \mathbf{a}) \tag{10}$$

In this case we can again apply the extremal control algorithm (8), but $i$-th component of vector $\mathbf{w}_l$ is calculated according to trial steps method

$$w_l^{(i)} = \frac{S(\mathbf{c}_n, \mathbf{a}_l - \delta_i) - S(\mathbf{c}_n, \mathbf{a}_l + \delta_i)}{2\sigma_i} \tag{11}$$

where $\delta_i$ is a zero vector except the $i$-th component equal to $\sigma_i$ (the trial step value).

Provided we have a fast enough systems simulator, it is also possible to make use of any of numerical optimization methods (including artificial intelligence - e.g.: reinforcement learning, genetic algorithms) to calculate optimal values of parameters $\mathbf{a}_l^*$ for a given systems parameters $\mathbf{C}_l$ by use of the following algorithm

$$\mathbf{a}_l^* = \arg\min_{\mathbf{a}} S_N(\mathbf{C}_l, \mathbf{a}). \tag{12}$$

Finally, simulator of the system may be applied as the training algorithm for the expert system $ES$, which would return optimal values of $\mathbf{a}_l^*$ for sequentially measured systems parameters $\mathbf{C}_l$

$$\mathbf{a}_l^* = ES(\mathbf{C}_l). \tag{13}$$

In this case, simulator is the trainer, and the learning expert may be based on one of the following approaches: knowledge base with a set of rules, neural network, or another expert with knowledge representation (e.g.: uncertain variables, fuzzy sets). Additionally, the trained expert may validate and update its knowledge during the run of the system [3].

The crucial issue that must be taken into consideration during the process of design of the adaptation system is the length of the adaptation step. It must be long enough to allow the computationally complex adaptation algorithm to execute. On the other hand, if the step is too long, systems conditions to which we are trying to adapt will change.

## 5. Simulation study

In order to show the difference between presented scheduling schemes a simple experimental example is provided. In this example three scheduling algorithms are compared: A1 - WRR, A2, A3 - WRR with adaptively changed weights. Algorithms A2 and A3 are based on the reinforcement learning approach. In A3 weighs are changed according to locally measured value of the quality of service criterion (output $\mathbf{y}_{m,n}$ for each algorithm $H_m$, $m = 1, \ldots, M+1$). Algorithm A2 uses additional information concerning the value of the global quality of service criterion $\mathbf{y}_n$.

Presented algorithms were evaluated according to the criterion function defined as follows

$$Q = (\sum_{k=1}^{K} p_k \cdot Q_k) \cdot (\sum_{k=1}^{K} p_k)^{-1} \tag{14}$$

where $p_k$ is the $k$-th class priority and $Q_k$ is $k$-th class quality of service criterion defined as

$$Q_k = (\sum_{t=0}^{T} \max\{0, Q_{t,k} - Q_k^*\}) \cdot (\sum_{t=0}^{T} Q_{t,k})^{-1} \tag{15}$$

where $T$ is the evaluation time horizon and $Q_k^*$ is $k$-th class QoS guarantee. Since $Q_k$ always satisfies $Q_k \in (0, 1)$, such a function has a natural interpretation as the percentage of overall QoS of traffic class k, that violated the quality of service guarantee $Q_k^*$.

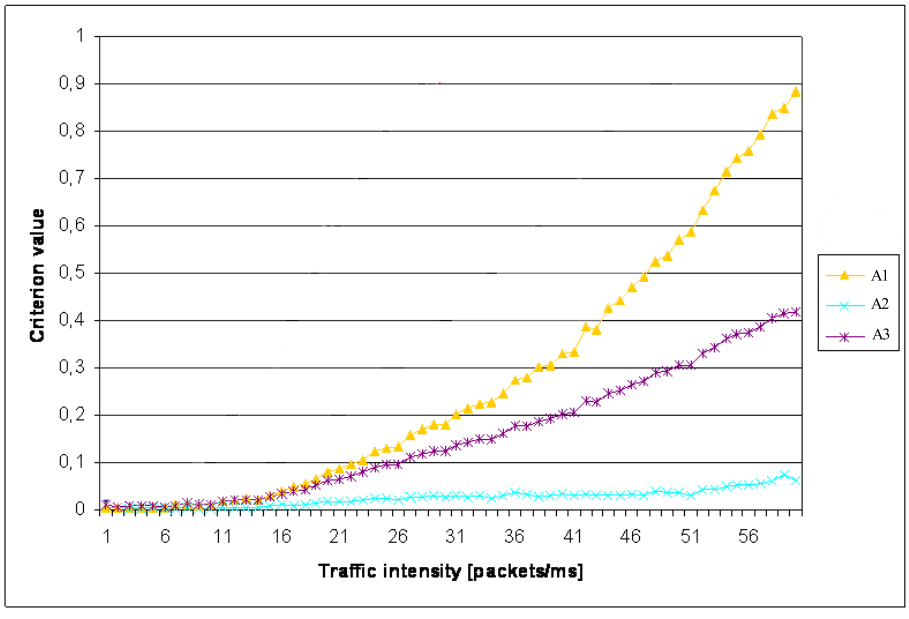



Fig. 7. Quality criterion function versus traffic intensity graph for three examined algorithms.

Figure 7 presents the quality of service versus traffic intensity graph for three evaluated algorithms. Simulation results confirmed that scheduling algorithms adapted basing on global quality criterion yield much better performance than the ones adapted basing on local criteria or not adapted at all.

## 6. Conclusions

In this paper a model of multistage packet scheduling system was introduced. Basing on the model three scheduling problems were formulated. Since defined problems belong to the class of NP-hard problems, exact algorithms cannot be applied in real-time systems. Therefore, it was shown how to adapt existing single stage scheduling algorithms to achieve high performance of the multistage packet processing system. Moreover, the discussion on possible heuristic on-line solution algorithms (including artificial intelligence methods) is provided. Finally, a simple simulation example, that justifies usefulness of proposed methods is given.

## References

[1] J.C.R. Bennett, H. Zhang: W$F^2$Q: Worst-case fair weighted fair queuing, in Proc. IEEE INFOCOM'96, San Francisco, CA, Mar. 1996, pp. 120-128.

[2] J.C.R. Bennett, H. Zhang: *Hierarchical packet fair queuing algorithms*, IEEE/ACM Trans. Networking, vol. 5, pp. 675-689, Oct. 1997.

[3] Z. Bubnicki: *Modern control theory*, Springer, Berlin, 2005

[4] W. Findeisen: *Struktury sterowania dla złożonych systemów*, Wydawnictwa Politechniki Warszawskiej, Warszawa, 1997

[5] S. Floyd, V. Jacobson: *Link-sharing and Resource Management Models for Packet Networks*, IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365-386, August 1995

[6] A. Grzech: *Sterowanie ruchem w sieciach teleinformatycznych*, Oficyna Wydawnicza Politechniki Wrocławskiej, 2002

[7] H. Sariowan, R.L. Cruz, G.C. Polyzos: *Scheduling for quality of service guarantees via service curves*. In Proceedings of the International Conference on Computer Communications and Networks (ICCCN) 1995, pages 512–520, September 1995

[8] D. Stiliadis, A. Varma: *Efficient fair queueing algorithms for packet-switched networks*, IEEE/ACM Trans. Networking vol. 6, no.2, pp. 175-185, 1998

[9] I. Stoica, H. Zhang, T.S.E. Ng: *A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time, and Priority Services*, IEEE/ACM Trans. Networking, vol. 8, no. 2, pp.185-199, APRIL 2000

[10] P. Świątek: *Providing Quality of Service and Network Security Policy in Computer Networks*, In Proceedings of the 16th International Conference of System Science, vol. II, pp. 337-346, September 2007, Wrocław

[11] A.S. Tanenbaum: *Computer Networks*, 3rd Edition, Prentice-Hall, 1996

**Acknowledgements**

This work was supported by the Polish State Committee for Scientific Research under Grant No. 3 T11C 029 29 (2005-2007).

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 235–246

# An Application of Lagrangean Decomposition to Optimization of Inter-Domain Routing in IP/MPLS Networks

MARISUZ MYCEK [a] ARTUR TOMASZEWSKI [b]

[a]Institute of Telecommunications
Warsaw University of Technology
mariusz@tele.pw.edu.pl

[b]Institute of Telecommunications
Warsaw University of Technology
artur@tele.pw.edu.pl

**Abstract:** 1 The goal of the paper is to present a decomposed approach to optimization of inter-domain routing in IP networks. A problem of maximizing the total amount of traffic carried in an inter-domain network is presented as a linear programme. Using Lagrangean relaxation the problem is decomposed with respect to individual domains. A subgradient optimization method for resolution of the problem combined with recovering of a near optimal primal solution is discussed, and its efficiency is compared with state of the art proximal bundle algorithm implemented in [21].

## 1. Introduction

Each domain of the Internet acts as Autonomous System and does reveal only a very limited information about its internal topology and implemented routing scheme to adjacent domains by means of exterior gateway protocols (EGP) such as BGP (see [1] and the discussion there). As analogous information it gathers from other domains is also fragmentary – a domain has only partial knowledge of the overall network topology what prevents it from making optimal (in a global sense) inter-domain routing decisions.

It seems that reaching a globally "optimal" traffic routing in the inherently decentralized Internet environment requires implementation of a distributed, network-wide process of routing optimization run in the control plane of the network. Some preliminary results on distributed inter-domain routing optimization can be found for example in [2], [3], [4], [5] and [6]. In [4] a generic multi-domain routing problem (consisting in optimization of bandwidth reservation levels on inter-domain links for traffic flows identified by traffic classes and traffic destinations) is formulated, and its possible decompositions are discussed. In [5] it is shown how to decompose the problem with respect to individual domains using sub-gradient

optimization based on Lagrangean relaxation. In [6] it is demonstrated how to resolve an inter-domain routing optimization problem using a distributed process based on sub-gradient optimization combined with recovering of near-optimal bandwidth reservation levels.

The paper is organized as follows. Section 2. reminds a formulation of the problem (as stated in [4]). Section 3. discusses an application of subgradient optimization method with a recovery of primal solution. Section 4. presents results of numerical experiments comparing effectiveness of subgradient method discussed in section 3. and advanced, state of the art, proximal bundle method (called ConicBundle 0.1) implemented in [21]. Eventually, Section 5. gives a short summary.

## 2. Problem formulation

Generally speaking, we consider a problem of maximizing the total amount of traffic carried in a multi-domain network (see [4]). The considered model of the network consists of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set of nodes $\mathcal{V}$ and the set of directed links $\mathcal{E}$ ($\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$). For a set of nodes $\mathcal{U} \subseteq \mathcal{V}$ we define the set $\delta^+(\mathcal{U})$ of links outgoing from set $\mathcal{U}$, and the set $\delta^-(\mathcal{U})$ of links incoming to set $\mathcal{U}$. More precisely, $\delta^+(\mathcal{U}) = \{e \in \mathcal{E} : a(e) \in \mathcal{U} \wedge b(e) \notin \mathcal{U}\}$ and $\delta^-(\mathcal{U}) = \{e \in \mathcal{E} : b(e) \in \mathcal{U} \wedge a(e) \notin \mathcal{U}\}$, where $a(e)$ and $b(e)$ denote the originating and terminating node, respectively, of link $e \in \mathcal{E}$. Besides, we shall write $\delta^{\pm}(v)$ instead of $\delta^{\pm}(\{v\})$, i.e., when $\mathcal{U} = \{v\}$ is a singleton.

$\mathcal{M}$ is the set of network domains. Each node $v \in \mathcal{V}$ belongs to exactly one domain denoted by $\mathcal{A}(v)$. Hence, set $\mathcal{V}$ is partitioned into subsets $\mathcal{V}^m = \{v \in \mathcal{V} : \mathcal{A}(v) = m\}, m \in \mathcal{M}$. For each domain $m \in \mathcal{M}$, $\mathcal{E}^m = \{e \in \mathcal{E} : a(e), b(e) \in \mathcal{V}^m\}$ is the set of *intra-domain* links between the nodes in the same domain $m$. The set of all intra-domain links is denoted by $\mathcal{E}_{\mathcal{I}} = \bigcup_{m \in \mathcal{M}} \mathcal{E}^m$. Further, the set of all *inter-domain* links is denoted by $\mathcal{E}_{\mathcal{O}}$, where $\mathcal{E}_{\mathcal{O}} = \{e \in \mathcal{E} : \mathcal{A}(a(e)) \neq \mathcal{A}(b(e))\} = \bigcup_{m \in \mathcal{M}} \delta^+(\mathcal{V}^m) = \bigcup_{m \in \mathcal{M}} \delta^-(\mathcal{V}^m)$. Clearly, the set of intra-domain links is disjoint with the set of inter-domain links. Finally, the capacity of link $e \in \mathcal{E}$ is denoted by $c_e$ and expressed in units of bandwidth, for example in Mb/s.

Set $\mathcal{D}$ represents traffic demands between pairs of nodes (not necessarily between all pairs). The originating and terminating node of demand $d \in \mathcal{D}$ is denoted by $s(d)$ and $t(d)$, respectively, and $h_d$ is the traffic volume of demand $d$, expressed in the same units of bandwidth as capacity of links. Also, $\mathcal{D}(s,t) = \{d \in \mathcal{D} : s(d) = s \wedge t(d) = t\}$ denotes the set of all demands from node $s \in \mathcal{D}$ to node $t \in \mathcal{D}$ (note that there can be more than one demand between a given pair on nodes). In the sequel, $z_d$ will denote the variable specifying the percentage of volume $h_d$ actually handled in the network, i.e., $z_d h_d$ is the carried traffic of demand $d$. The set of all demands originating in domain $m$ is denoted as $\mathcal{D}^m = \{d \in \mathcal{D} : s(d) \in \mathcal{V}^m\}$. The sets $\mathcal{D}^m = \{d \in \mathcal{D} : s(d) \in \mathcal{V}^m\}$, $m \in \mathcal{M}$, define a partition of $\mathcal{D}$.

Let $x_{et}$ denote a variable specifying the amount of aggregated bandwidth (called *flow* in the sequel) reserved on intra-domain link $e \in \mathcal{E}_{\mathcal{I}}$ for the traffic destined for (a remote) node $t \in \mathcal{V}$. Then, for each inter-domain link $e \in \mathcal{E}_{\mathcal{O}}$ we introduce two flow variables: $x^+_{et}$ and $x^-_{et}$.

Variable $x^+_{et}$ (respectively, $x^-_{et}$) denotes the amount of bandwidth reserved for traffic carried on $e$ and destined for $t$ that is reserved by domain $\mathcal{A}(a(e))$ (respectively, $\mathcal{A}(b(e))$) at which link $e$ originates (respectively, terminates). Then for each domain $m \in \mathcal{M}$ we introduce the following flow vectors:

- $\boldsymbol{z}^m = (z_d : d \in \mathcal{D}^m)$
- $\boldsymbol{x}^m = (x_{et} : e \in \mathcal{E}^m, t \in \mathcal{V})$
- $\boldsymbol{x}^{m+} = (x^+_{et} : e \in \delta^+(\mathcal{V}^m), t \in \mathcal{V})$
- $\boldsymbol{x}^{m-} = (x^-_{et} : e \in \delta^-(\mathcal{V}^m), t \in \mathcal{V})$
- $\boldsymbol{X}^m = (\boldsymbol{z}^m, \boldsymbol{x}^m, \boldsymbol{x}^{m+}, \boldsymbol{x}^{m-})$.

The basic conditions that have to be fulfilled in each domain $m \in \mathcal{M}$ are flow conservation constraints

$$\begin{aligned} &\sum_{e\in\delta^+(v)\cap\mathcal{E}^m} x_{et} + \sum_{e\in\delta^+(v)\setminus\mathcal{E}^m} x^+_{et} \\ -&\sum_{e\in\delta^-(v)\cap\mathcal{E}^m} x_{et} - \sum_{e\in\delta^-(v)\setminus\mathcal{E}^m} x^-_{et} \\ &\qquad = \sum_{d\in\mathcal{D}(v,t)} z_d h_d, \quad t \in \mathcal{V}, v \in \mathcal{V}^m \setminus \{t\} \end{aligned} \tag{1a}$$

and capacity constraints

$$\sum_{t\in\mathcal{V}} x_{et} \le c_e, \qquad e \in \mathcal{E}^m \tag{1b}$$

$$\sum_{t\in\mathcal{V}} x^+_{et} \le c_e, \qquad e \in \delta^+(\mathcal{V}^m) \tag{1c}$$

$$\sum_{t\in\mathcal{V}} x^-_{et} \le c_e, \qquad e \in \delta^-(\mathcal{V}^m). \tag{1d}$$

Let $\boldsymbol{\mathcal{X}}^m$ ($m \in \mathcal{M}$) denote the set of all vectors $\boldsymbol{X}^m$ satisfying constraints (1) and, possibly, certain extra domain-specific conditions. Such extra constraints can for example be implied by requirements for the weight-based shortest-path intra-domain routing (see Chapter 7 in [8]) or by QoS-type conditions such as $z_d \ge 1, d \in \mathcal{D}^m$.

The routing optimization problem can now be stated as follows:

$$\max \; F(\boldsymbol{z}) = \sum_{m\in\mathcal{M}} \sum_{d\in\mathcal{D}^m} z_d h_d \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{X}^m \in \boldsymbol{\mathcal{X}}^m, \qquad m \in \mathcal{M} \tag{2b}$$

$$x^+_{et} \le x^-_{et}, \qquad e \in \mathcal{E}_{\mathcal{O}},\ t \in \mathcal{V}. \tag{2c}$$

Certainly, objective functions different from (2a) can also be considered.

Let $\boldsymbol{\lambda} = (\lambda_{et} : e \in \mathcal{E}_{\mathcal{O}}, t \in \mathcal{V})$ be a vector of (non-negative) multipliers associated with constraints (2c). As shown in [4], the Lagrangean function $L(\boldsymbol{\lambda}; \boldsymbol{X}), \boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{X} = (\boldsymbol{X}^m : m \in \mathcal{M}) \in \boldsymbol{\mathcal{X}} = \bigotimes_{m \in \mathcal{M}} \boldsymbol{\mathcal{X}}^m$ associated with problem (2) is of the following decomposed form:

$$L(\boldsymbol{\lambda}; \boldsymbol{X}) = \sum_{m \in \mathcal{M}} L^m(\boldsymbol{\lambda}^m; \boldsymbol{X}^m). \tag{3}$$

In (3), $\boldsymbol{\lambda}^m = (\lambda_{et} : e \in \delta^-(\mathcal{V}^m) \cup \delta^+(\mathcal{V}^m), t \in \mathcal{V})$ is the sub-vector of $\boldsymbol{\lambda}$ composed of the values $\lambda_{et}$ for all inter-domain links $e$ originating or terminating in domain $m \in \mathcal{M}$, and $L^m(\boldsymbol{\lambda}^m; \boldsymbol{X}^m)$ denotes the partial Lagrangean corresponding to domain $m \in \mathcal{M}$ equal to

$$\sum_{d \in \mathcal{D}^m} z_d h_d \; + \sum_{t \in \mathcal{V}} \Big( \sum_{e \in \delta^-(\mathcal{V}^m)} \lambda_{et} x^-_{et} - \sum_{e \in \delta^+(\mathcal{V}^m)} \lambda_{et} x^+_{et} \Big), \tag{4}$$

where $\boldsymbol{\lambda}^m \geq \mathbf{0}$ and $\boldsymbol{X}^m \in \boldsymbol{\mathcal{X}}^m$ .

The problem dual to (2) (see for example [9]) becomes as follows:

$$w^* = \min_{\boldsymbol{\lambda} \geq \mathbf{0}} w(\boldsymbol{\lambda}). \tag{5}$$

The (non-empty) set of optimal solutions of problem (5) will be denoted by $\boldsymbol{\Lambda}^*$. The dual function $w$ is defined as $w(\boldsymbol{\lambda}) = \sum_{m \in \mathcal{M}} w^m(\boldsymbol{\lambda}^m)$ and is computed through resolving separate subproblems:

$$w^m(\boldsymbol{\lambda}^m) = \max_{\boldsymbol{X}^m \in \boldsymbol{\mathcal{X}}^m} L^m(\boldsymbol{\lambda}^m; \boldsymbol{X}^m), \; m \in \mathcal{M}. \tag{6}$$

For any $\boldsymbol{\lambda} \geq \mathbf{0}$, $\boldsymbol{X}(\boldsymbol{\lambda}) \in \boldsymbol{\mathcal{X}}$ will denote the so called *maxi- mizer* of the Lagrangean function (3), i.e., any optimal solution of the Lagrange problem:

$$\boldsymbol{X}(\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{X} \in \boldsymbol{\mathcal{X}}} L(\boldsymbol{\lambda}; \boldsymbol{X}). \tag{7}$$

Any maximizer $\boldsymbol{X}(\boldsymbol{\lambda}) = (\boldsymbol{X}^m(\boldsymbol{\lambda}^m) : m \in \mathcal{M})$ is computed through solving independent subproblems (6):

$$\boldsymbol{X}^m(\boldsymbol{\lambda}^m) = \arg\max_{\boldsymbol{X}^m \in \boldsymbol{\mathcal{X}}^m} L^m(\boldsymbol{\lambda}^m; \boldsymbol{X}^m), \; m \in \mathcal{M}. \tag{8}$$

In the sequel the quantity $\nabla w(\boldsymbol{\lambda})$ will denote a subgradient of the dual function $w$ at point $\boldsymbol{\lambda}$. Subgradients are obtained as a by-product of the (distributed) computation of the values of $w(\boldsymbol{\lambda})$: if $\boldsymbol{X}(\boldsymbol{\lambda})$ is a maximizer of the Lagrangean function (3) for a given $\boldsymbol{\lambda}$, then the corresponding subgradient $\nabla w(\boldsymbol{\lambda})$ is as follows ([9]):

$$\nabla w(\boldsymbol{\lambda}) = (x(\boldsymbol{\lambda})^-_{et} - x(\boldsymbol{\lambda})^+_{et} : e \in \mathcal{E}_{\mathcal{O}}, t \in \mathcal{V}). \tag{9}$$

### 3. Subgradient minimization and recovery of primal solution

The dual problem can be resolved using general subgradient minimization (SM) techniques (see for example [10] and [11]), as explained in [4]. However, this is not sufficient because as discussed in [5] an optimal dual solution $\boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}^*$ of problem (5) does not in general yield an optimal (nor even feasible) primal solution $\boldsymbol{X}^*$, i.e., an optimal solution of problem (2). What we only know for sure is that any optimal $\boldsymbol{\lambda}^*$ gives the optimal value $F^*$ of the primal objective function (2a): $F^* = w(\boldsymbol{\lambda}^*) = w^*$. We note here that although any optimal primal solution $\boldsymbol{X}^*$ is a maximizer for any $\boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}^*$, the converse is not true. In fact, in general a maximizer $\boldsymbol{X}(\boldsymbol{\lambda}^*)$ can be primal infeasible.

Still, SM can be combined with recovering a (near-) optimal primal solution $X^*$, leading to a class of algorithms referred to as SM-PR (subgradient minimization—primal recovery). The idea, due to [12], has been elaborated in [13], and more recently in [14], [15], and [16]. In SM-PR algorithms an optimal primal solution is recovered from the sequence of maximizers $(\boldsymbol{X}(\boldsymbol{\lambda}_j) : j = 0, 1, \ldots, k)$ of the Lagrangean function (3) computed for the vectors of dual variables $\boldsymbol{\lambda}_j$ obtained in the consecutive steps $j = 0, 1, \ldots$ of the SM process. The crux is that a convex combination $\sum_{j=0}^{k} \mu_j(k) \boldsymbol{X}(\boldsymbol{\lambda}_j)$ (with suitably chosen $\mu_j(k) \geq 0,\ \sum_{j=0}^{k} \mu_j(k) = 1$) can be made a tight approximation of an optimal primal solution $\boldsymbol{X}^*$ of (2). A general form of an SM-PR algorithm is as follows.

**Step 0:** Set initial $\boldsymbol{\lambda}_0$ and initial step-size $\gamma_0$. Put $\boldsymbol{X}_0^* = \boldsymbol{X}(\boldsymbol{\lambda}_0)$ and set the iteration counter $k$ to 0.
**Step 1:** Put $k := k + 1$ and:
**(a)** determine step-size $\gamma_k$; $\boldsymbol{\lambda}_k := (\boldsymbol{\lambda}_{k-1} - \gamma_k \nabla w(\boldsymbol{\lambda}_{k-1}))_+$
**(b)** determine coefficients $\mu_j(k)$; $\boldsymbol{X}_k^* := \sum_{j=0}^{k} \mu_j(k) \boldsymbol{X}(\boldsymbol{\lambda}_j)$.
**Step 2:** If the stopping criterion not satisfied go to Step 1.

In Step 1(a), vector $(t_1, t_2, \ldots, t_N)_+$ denotes the projection vector $(\max\{t_1, 0\}, \max\{t_2, 0\}, \ldots, \max\{t_N, 0\})$, used for projecting the current point onto the feasible region $\boldsymbol{\lambda} \geq \mathbf{0}$. The above algorithm will converge to a dual optimal solution $\boldsymbol{\lambda}^* = \lim_{k\to\infty} \boldsymbol{\lambda}_k$, and a primal optimal solution $\boldsymbol{X}^* = \lim_{k\to\infty} \boldsymbol{X}_k^*$, provided that the step-size sequence $(\gamma_k)_{k\geq 0}$ and the related sequence of coefficients $(\mu_j(k) : j = 0, 1, \ldots, k)_{k\geq 0}$ are properly constructed.

There are several ways of constructing the sequences in question that can be found in the literature. In [12] the step-size sequence is any sequence such that

$$\forall k \geq 0,\ \gamma_k > 0, \qquad \lim_{k\to\infty} \gamma_k = 0, \qquad \sum_{k=0}^{\infty} \gamma_k = \infty, \tag{10}$$

and the corresponding coefficients of the $\boldsymbol{X}^*$-defining combinations are given by

$$\forall k \geq 0,\ \mu_j(k) = \frac{\gamma_j}{\sum_{j=0}^{k} \gamma_j},\ j = 0, 1, \ldots, k. \tag{11}$$

In [14] the step-sizes form a harmonic sequence

$$\gamma_k = \frac{a}{b+ck}, \quad k = 1, 2, \ldots \tag{12}$$

(with appropriate constants $a, c > 0$ and $b \geq 0$; for example $a = c = 1, b = 0$ implying $\gamma_k = \frac{1}{k}$) which makes it possible to use coefficients $\mu_j(k)$ equal to each other:

$$\forall k \geq 0,\ \mu_j(k) = \frac{1}{k+1},\ j = 0, 1, \ldots, k. \tag{13}$$

This gives a convenient formula for Step 1(b):

$$\boldsymbol{X}_k^* := \frac{\sum_{j=0}^{k} \boldsymbol{X}(\boldsymbol{\lambda}_j)}{k+1}. \tag{14}$$

In [16] the step size is defined (following [17]) as

$$\gamma_k = \eta \frac{\tilde{w}_{k-1} - \hat{w}}{\|\nabla w(\boldsymbol{\lambda}_{k-1})\|^2}, \tag{15}$$

where $0 < \eta \leq 2$ is a fixed number, $\|\cdot\|$ is the Euclidean norm, $\tilde{w}_{k-1} = \min_{0 \leq j \leq k-1} w(\boldsymbol{\lambda}_j)$, and $\hat{w}$ is a lower bound of the dual objective $w^*$. (Observe that $\nabla w(\boldsymbol{\lambda}_{k-1}) = \mathbf{0}$ means that the optimum $w^*$ has been reached.) The corresponding coefficients $\mu_j(k)$ are defined by a geometric-like series, i.e.,

$$\mu_0(k) = (1-\alpha)^k, \quad \mu_j(k) = \alpha(1-\alpha)^{k-j},\ j = 1, 2, \ldots, k, \tag{16}$$

where $0 < \alpha < 1$. This leads to the following convenient formula for Step 1(b):

$$\boldsymbol{X}_k^* := \alpha \boldsymbol{X}(\boldsymbol{\lambda}_k) + (1-\alpha)\boldsymbol{X}_{k-1}^*. \tag{17}$$

Because of the third property in (10), the above SM-PR methods are sometimes called *divergent series* methods. For a more rigid discussion on the divergent series methods the reader is referred to [15].

It turns out that convergence of the divergent series SM-PR algorithms can be slow even when applied to medium-size instances of (2). Certain improvement can be achieved when using deflected subgradients in Step 1(a) (see Section 3 in [15]). Still, it seems that one of the best available SM-PR methods are the so called proximal bundle methods, see [18], [19], and [20]. Therefore, in our numerical results discussed in Section 4. we also used a SM-PR algorithm of the proximal bundle type (called ConicBundle 0.1) implemented in [21] (for a brief theoretical introduction see Section 5 in [22]).

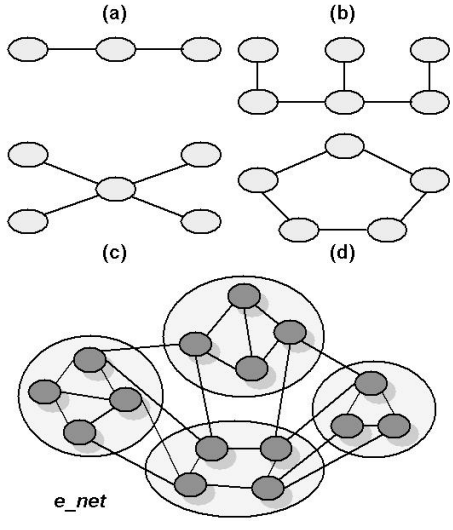



Fig. 1. Inter-domain network structures

## 4. Numerical experiments

Most of the networks used in our numerical experiments are of one from four inter-domain connectivity types (a)-(d) depicted in Figure 1 (where circles represent domains, and lines represent groups of links between the domains). For each type of connectivity three variants are considered, with one, two, and three fully connected nodes in each domain. The resulting network instances are referred to as "x_n", where x denotes one of the domain connectivity types and n denotes the number of nodes in each domain.

We also studied two larger network examples: *e_net* depicted in Figure 1, and a 7-domain network *r_net* with 40 nodes, 184 directed links and 1560 directed demands. In both cases, adjacent nodes are connected by two oppositely directed arcs of the same capacity, and a non-zero demand is assumed between each pair of nodes.

Below (in Tables 1 and 2), we present results for two SM-PR algorithms based on two approaches discussed in Section 3. for simultaneous resolution of the dual problem (5) and recovery of an optimal solution of the primal problem (2):

- SG: an algorithm based on the divergent series method
- CB: an algorithm using an advanced library for minimization of a piece-wise linear function and aggregation of primal solutions implemented in [21].

.

In these two experiments we assumed the range ($z_d \geq 1$) for variables $z_d$ (see (2a) and discussion after formula (1)). We investigated the overall quality of the primal solution produced by the algorithms as well as how this quality depends on the accuracy assumed for

the resolution of the dual problem. The accuracy of the dual solution is denoted by $\beta = \frac{\tilde{w}}{F^*}$, where $F^*$ denotes the value of the optimal primal objective (computed directly by applying the CPLEX solver to formulation (2)) and $\tilde{w}$ denotes the value of the dual solution (an upper bound on $F^*$).

Let $\mathcal{D}_\mathcal{O} = \{d \in \mathcal{D} : \mathcal{A}(s(d)) \neq \mathcal{A}(t(d))\}$ be the set of all inter-domain demands, and let $\overline{h} = \frac{\sum_{d \in \mathcal{D}_\mathcal{O}} h_d}{|\mathcal{D}_\mathcal{O}|}$ be the average volume of an inter-domain demand. Further, let $\mathcal{Q} = \{(e,t) : e \in \mathcal{E}_\mathcal{O}, t \in \mathcal{V}, x^+_{et} > 0\}$ be the set of all pairs $(e,t)$ with non zero inter-domain reservation level $x^+_{et}$, and let $\mathcal{R} = \{(e,t) \in \mathcal{Q}, x^+_{et} > x^-_{et} + \varepsilon\overline{h}\}$ be the set of all pairs $(e,t)$ with primal infeasible reservation levels $x^+_{et}$ and $x^-_{et}$. The quantity $\varepsilon$ is a small positive constant used to take care of numerical inaccuracies (and, potentially, for intrinsic uncertainty of demand volumes measurements). In our experiments, we assumed $\varepsilon = 0.01$.

The quality of a solution is described by parameters $P$, $N$[%], $V$[%] and $T$[s]. $P$ is equal to the objective value of the recovered primal solution divided by the known optimal primal objective value $F^*$, $N = 100\frac{|\mathcal{R}|}{|\mathcal{Q}|}$ denotes the percentage of pairs $(e,t)$ that violate constraint (2c), $V = 100\frac{\sum_{(e,t)\in\mathcal{R}} (x^+_{et} - x^-_{et})}{\overline{h}|\mathcal{R}|}$ is the relative mean value of the violation, and $T$ denotes the computation time.

In our implementation of the divergent series SM-PR algorithm step-sizes $\gamma_k$ are computed according to (15), and the coefficients $\mu_j(k)$ according to (16). As the lower bound value $\hat{w}$ we take the exact value $w^* = F^*$ computed directly for formulation (2). The algorithm starts with the step-size modifier $\eta = 2$ and halves this value every time there has not been sufficient improvement in the dual objective (in our case—one thousandth of the optimal primal objective value) during consecutive 32 iterations. The convex combination weights for the maximizers of the Lagrangean function are taken into account through (17). The algorithm stops when the assumed accuracy $\beta$ is reached, or the assumed maximum number of iterations has been exceeded.

All the computations were performed on an Intel class computer with a 3.0 GHz Pentium-4 processor and 1 GB of RAM. All linear subproblems were solved using CPLEX 10.1.

The first observation is that SG (cf. Table 1) performs rather poorly. Although in each case the value of the primal objective for the recovered primal solution is pretty good, still the percentage of violated inter-domain reservation levels $N$ and the average magnitude of this violation $V$ are unacceptable. In most cases we were not able to obtain solutions for small values of $\beta$ (as the algorithm stops earlier). However, even in the cases when we could achieve higher accuracy no significant improvement in terms of $N$ and $V$ was possible to achieve. It is worth mentioning that achieving accuracy of $\beta = 1.01$ takes from 5 up to 10 times more computation time than it is required for accuracy $\beta = 1.05$.

The second observation is that CB (cf. Table 2) performs excellent, except for the case of

| | $\beta = 1.1$ | | | | $\beta = 1.05$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $P$ | $N$ | $V$ | $T$ | $P$ | $N$ | $V$ | $T$ |
| a_1 | .88 | 32 | 31 | 11 | .89 | 40 | 20 | 29 |
| b_1 | 1.0 | 36 | 17 | 5 | 1.0 | 38 | 15 | 9 |
| c_1 | .88 | 40 | 6.6 | 23 | .88 | 40 | 6.5 | 36 |
| d_1 | .86 | 23 | 25 | 14 | .86 | 37 | 32 | 106 |
| a_2 | .98 | 37 | 51 | 180 | .94 | 34 | 51 | 235 |
| b_2 | .99 | 35 | 37 | 154 | 1.0 | 48 | 33 | 300 |
| c_2 | .91 | 33 | 29 | 206 | .87 | 50 | 23 | 571 |
| d_2 | .88 | 53 | 89 | 943 | .82 | 39 | 23 | 1080 |
| a_3 | .95 | 48 | 61 | 1320 | .96 | 39 | 88 | 1524 |
| b_3 | .96 | 53 | 59 | 1039 | .96 | 42 | 77 | 1144 |
| c_3 | .95 | 45 | 58 | 702 | .98 | 49 | 43 | 941 |
| d_3 | .95 | 41 | 34 | 675 | .93 | 38 | 51 | 790 |
| e_net | .97 | 46 | 15 | 1094 | .99 | 44 | 18 | 1548 |
| r_net | | - | | - | - | - | - | - |

Table 1. SG Numerical results for $\boldsymbol{z} \geq 1$

| | $\beta = 1.1$ | | | | $\beta = 1.05$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $P$ | $N$ | $V$ | $T$ | $P$ | $N$ | $V$ | $T$ |
| a_1 | .88 | 29 | 9.4 | 1 | .88 | 16 | 4.8 | 2 |
| b_1 | 1.0 | 24 | 30 | 1 | 1.0 | 30 | 13 | 1 |
| c_1 | .88 | 0.0 | 0.0 | 2 | .90 | 0.0 | 0.0 | 3 |
| d_1 | .86 | 13 | 5.0 | 1 | .86 | 0.0 | 0.0 | 2 |
| a_2 | .90 | 32 | 11 | 40 | .91 | 31 | 6.0 | 72 |
| b_2 | .90 | 27 | 11 | 29 | .90 | 19 | 4.7 | 45 |
| c_2 | .89 | 25 | 7.2 | 33 | .89 | 3.3 | 1.6 | 77 |
| d_2 | .81 | 14 | 4.0 | 36 | .81 | 5.8 | 3.1 | 58 |
| a_3 | .99 | 42 | 17 | 187 | .99 | 42 | 11 | 270 |
| b_3 | 1.0 | 41 | 23 | 202 | .99 | 43 | 11 | 258 |
| c_3 | .99 | 42 | 28 | 196 | .98 | 39 | 16 | 235 |
| d_3 | .96 | 40 | 16 | 121 | .96 | 37 | 6.9 | 210 |
| e_net | .92 | 36 | 19 | 64 | .94 | 36 | 9.8 | 121 |
| r_net | .99 | 46 | 29 | 7020 | 1.0 | 40 | 14 | 12000 |

Table 2. CB Numerical results for $\boldsymbol{z} \geq 1$

the large network *r_net*, for which we could not obtain any solution in one of the experiments. In each case, the recovered primal solution is near-optimal, the percentage $P$ of violated inter-domain reservation levels is moderate and, what is the most important, the scale of violation $V$ is very small. Basically, resolving the dual problem with accuracy $\beta = 1.05$ gives a good quality primal solution in a reasonable time. As in the case of SG, increasing the required accuracy to the level of $\beta = 1.01$ considerably increases (5-10 times) the computation time, but in the case of CB the remaining quality parameters $P, N, V$ are at the same time getting significantly better.

We may conclude our observations saying that the CB algorithm is very promising both in terms of acceptable quality of the recovered primal solutions and the computation time.

## 5. Concluding remarks

Because of a highly distributed nature of the today's Internet, and because the operators are not willing to disclose sensitive information concerning their domains, any process of inter-domain traffic routing optimization must be distributed across the domains and based on parameters exchanged between neighboring domains by means of EGP protocols.

In the paper we have presented a global multi-domain routing design problem (2) consisting in optimization of bandwidth reservation levels on inter-domain links for traffic flows identified by traffic classes and traffic destinations. We have shown how to decompose the problem with respect to individual domains using Lagrangean relaxation, and demonstrated how to resolve the problem using subgradient optimization combined with recovering of near-optimal bandwidth reservation levels.

Despite some interesting initial trials (see [3], [2] for the two-domain case), an effective, distributed optimization processes for inter-domain routing optimization is yet to be found. We believe that the presented paper shows a realistic way towards defining such an implementable process. This would be an automatic process running in the control plane of the network, continuously solving problem (2) in real time, and adapting to changing traffic and link availability conditions. The whole process would be distributed across the set of network domains; each domain would be responsible for optimizing its own routing, and for computing the piece of inter-domain information it is responsible for. Only a limited cooperation between domains would be needed; each domain would be required to exchange with its neighbors the information specifying $\boldsymbol{x}^{m-}$ and $\boldsymbol{x}^{m+}$ (and to agree on $\boldsymbol{\lambda}^{m+}$), using appropriate TE extensions of EGP protocols.

However, a fundamental issue is still to be solved: to what extent and how to synchronize the sub-processes running in individual domains to make the global process scalable and effective, both in terms of convergence, and the quality of bandwidth reservation levels with respect to the true optimum. Defining such a network-wide distributed optimization process will be a subject of future work.

**Acknowledgment.** The presented work was sponsored by Polish Ministry of Science and Higher Education (grant 3 T11D 001 27).

## References


[1] N. Feamster, J. Borkenhagen, and J. Rexford, Guidelines for Interdomain Traffic Engineering, *ACM SIGCOM Computer Communications Review*, vol.33, no.5, pp.19–30, October 2003.

[2] J. Winnick, S. Jamin, and J. Rexford, *Trafffic Engineering Between Neighboring Domains*, Technical Report, July 2002 (http://www.cs. princeton.edu/∼jrex/publications.html).

[3] G. Shrimali, A. Akella, and A. Mutapcic, *Cooperative Inter-Domain Traffic Engineering Using Nash Bargaining and Decomposition*, accepted for *IEEE INFOCOM'2007*, 2007 (http://www.stanford.edu/ ∼gireesh/traffic_engg_infocom_submit.pdf).

[4] A. Tomaszewski, M. Pióro et al., *Towards Distributed Inter-Domain Routing Optimization for IP/MPLS Networks*, Technical Report, Warsaw University of Technology, 2007, http://ztit.tele.pw.edu.pl /TR/NDG/rmd07.pdf

[5] M. Pióro, A. Tomaszewski et al., A Subgradient Optimization Approach to Inter-domain Routing in IP/MPLS Networks, *Proc. Networking'2007*, 2007.

[6] M. Pióro, A. Tomaszewski, and M. Mycek, A Distributed Scheme for Inter-Domain Routing Optimization, *Proc. DRCN'2007*, 2007.

[7] M. Pióro, A. Tomaszewski, and M. Mycek, Distributed Inter-Domain Link Capacity Optimization for Inter-Domain IP/MPLS Routing, *Proc. Globecom'2007*, 2007.

[8] M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*, Morgan Kaufman, 2004.

[9] L. Lasdon, *Optimization Theory for Large Systems*, MacMillan, 1970.

[10] M. Minoux, *Mathematical Programming: Theory and Algorithms*, John Wiley & Sons, 1986.

[11] J.E. Shapiro, *Mathematical Programming: Structures and Algorithms*, John Wiley & Sons, 1979.

[12] N. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, 1985.

[13] T. Larsson and Z. Liu, *A Primal Convergence Result for Dual Subgradient Optimization with Application to Multicommodity Network Flows*, Research Report, Department of Mathematics, Linköping Institute of Technology, Sweden, 1989.

[14] T. Larsson and Z. Liu, *A Lagrangean Relaxation Scheme for Structured Linear Programs with Application to Multicommodity Network Flows*, Research Report, Department of Mathematics, Linköping Institute of Technology, Sweden, 2004.

[15] H. Sherali and G. Choi, Recovery of Primal Solutions when Using Subgradient Optimization Methods to Solve Lagrangean Duals of Linear Programs, *Operations Research Letters*, vol.19, pp.105–113, 1996.

[16] F. Barahona and R. Anbil, The Volume Algorithm: Producing Primal Solutions with a Subgradient Method, *Journal of Mathematical Programming*, vol.87, no.3, pp.385–399, 2000.

[17] M. Held, P. Wolfe, and H. Crowder, Validation of Sub-Gradient Optimization, Mathematical Programming, vol.6, pp.62–88, 1974.

[18] K.C. Kiwiel, Approximations in Proximal Bundle Methods and Decomposition of Convex Programs, *J. Optim. Theory Appl.*, vol.84, pp.529-548, 1995.

[19] J-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization I, II*, Springer-Verlag, 1996.

[20] S. Feltenmark and K.C. Kiwiel, Dual Applications of Proximal Bundle Methods, Including Lagrangian Relaxation of Nonconvex Problems, *SIOPT*, vol.10, no.3, pp.697–721, 2000.

[21] C. Helmberg, *ConicBundle 0.1*, Fakultät für Mathematik, Technische Universität Chemnitz, 2005 (http://www-user.tu-chemnitz.de/ ∼helmberg/ConicBundle/).

[22] C. Helmberg and S. Róhl, A Case Study of Joint Online Truck Scheduling and Inventory Management for Multiple Warehouses, to appear in *Operations Research*.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 247–258

# Efficient path generation in resilient single backup routing optimization*

MATEUSZ DZIDA[a]

[a]Institute of Telecommunications, Warsaw University of Technology
Polish Telecom R&D
mdzida@tele.pw.edu.pl

**Abstract:** In this paper we address the problem of link dimensioning in resilient networks allowing traffic bifurcation in the nominal network state and restoration of each nominal flow on a single backup path. Considered problem consists in the selection of path pairs: nominal/backup that can be used to satisfy given traffic demands, so the cost of link capacities is the smallest. Assumed network failure model admits only link failures, that in general case concern bundle of links. Failed links became into the state of temporary unavailability due to the transmission break-down. Considered problem is formulated as a Linear Programming (LP) problem in the link-path (L-P) notation of multi-commodity flow optimization. However, the L-P notation seems useful only when effective column generation scheme is proposed. Thus, we discuss in the paper the applicability of this method on the basis of the theory of duality of linear programming. Since the column generation is in this case NP-complete, the paper presents two approaches of simplified efficient column generation and discuss usefulness of these approaches against exact column generation demanding solving of a Mixed Integer Programming (MIP) problem.

**Keywords:** : path generation, resilient routing optimization, link dimensioning, single backup protection

## 1. Introduction

Enormous and growing capacities of the transmission systems in nowadays telecommunication networks, may drastically influence the network performance in case of the system failure. Increasing aggregation level of the traffic flows makes them more and more sensitive on the network failures. For this reason, transport network operators are interested in providing reliable services, that thanks to fast recovery mechanisms are able to survive network failures. So, to provide reliable transport services, the transport network operator must foresee all failures that can appear in the network and accordingly to them design protection paths that can absorb the traffic overflows from failed nominal paths. In this paper, we investigate

*The research presented in this paper has been sponsored by Polish State Committee for Scientific Research (grant N N517 4395 33: "Efektywne metody projektowania tras kierowania ruchu najkrótszą ścieżką w sieci Internet odpornej na awarie") and Warsaw University of Technology (grant: "Opracowanie modeli i algorytmów optymalizacji sieci teletransmisyjnych z uwzględnieniem protekcji zasobów").

the link dimensioning of resilient routing demanding use of single backup paths to protect nominal flows against any provisioned failure. Hence, the problem consists in searching for a configuration of path pairs composed of nominal and dedicated backup paths, such that cost of link capacities resulting from realization of the given traffic demands is the smallest. Observe, that at least one of the paths creating a path pair must be always available, i.e., any failure cannot touche both paths from the pair at the same time. If it is not so, the corresponding demand would be not able to survive failures touching both paths from the pair. Considered problem is well recognized in the literature and it was considered in a series of research papers, i.e., [10], [1], [4], and [2].

In this paper we assume that although each nominal flow must be protected by a single backup path, any traffic bifurcation on nominal paths is allowed. Thus, the problem can be modelled in such a way that to each path pair there is assigned a flow variable and total demand volume is realized by a configuration of such path pairs with assigned bandwidth volumes. If a nominal path became unavailable appropriate traffic portion is transmitted through backup path.

Transport network operator interested in solving the problem of link dimensioning in resilient networks may consider use of optimization techniques. As far as the compact node-link formulation of the considered problem has not been proposed, the general problem formulation demands use of the L-P notation. However, this notation seems useful only in two situations:

1. when all path pairs can be predefined,
2. when a path pair generation method, defined upon the column generation technique, can be proposed.

Since, the first approach is not scalable with growing number of nodes, we focus on the second method. Path generation is a technique evolving from the general column generation technique, developed to effectively solve large-scale LP problems of special structures. This technique is based on the generation of candidate paths during the problem solving. General path generation framework can be described as follows. Having given traffic demands and costs of the link capacities, we determine preliminary sets of the candidate path pairs, containing at least one nominal path together with corresponding backup path for each demand. Using defined sets of candidate path pairs we can formulate the LP problem in the L-P notation. Having the LP problem solved (using LP solvers like CPLEX, X-PRESS), we can get the values of the dual Lagrangean multipliers related to the constraints of the LP problem. On the basis of the duality theory of linear programming, we can conclude (from values of these multipliers) if determined sets of candidate path pairs should be extended or not to get the optimal flow distribution in the sense of minimal capacity cost. If a set of candidate path pairs for a specific demand does not contain a pair that due to dual constraints may be useful, so-called *pricing problem* is formulated and the required pair can be obtained as a solution of this problem. The procedure consists in re-optimizing LP problem introducing

new path pairs, generated by pricing problem, until no pair out of candidate lists satisfies dual constraints and therefore the optimal solution is reached.

Application of the path generation technique allows to keep the lists of candidate path pairs very short comparing to the sets of all possible pairs. Thanks to that, the corresponding LP problem can be solved very efficiently. Still, the efficiency of the overall technique depends on the efficiency of path pair generation methods (pricing problem.) In the case of simple routing optimization problems (nominal state dimensioning), the generation may be very efficient due to the possibility of solving a pricing problem using a dynamic programming (shortest path algorithm.) Unfortunately, an efficient pair generation method was not proposed yet, because of complex structure of the pricing problem in case of resilient routing optimization. It appears that, generated nominal paths are conditional shortest paths depending on using selected backup paths. As you can easily see, generation of path pairs (nominal and backup) is therefore not trivial even for the starting sets of candidate path pairs. Assuming single link failures scenario such pairs may be generated as the shortest link-disjoint pairs (see [3], [11], and [12]), but in the general case, the problem of pair generation is proved to be NP-complete (see [8] and [4]). However, the pricing problem always can be formulated as a MIP problem.

As it was mentioned, generation of path pairs is not trivial and in the general case is NP-complete. For this reason, the application of the exact generation methods may seem not efficient and sometimes it may be satisfactory to obtain just approximated solutions. Therefore, in this paper we propose two heuristic methods of solving the considered problem. Application of these methods makes the generation quite easy and allows to obtain good quality solutions of the overall problem. Both discussed methods are based on the assumption that nominal paths are not generated during the problem solving at all. Instead of that, they are determined in preliminaries as solutions of the k-shortest path problem or as solution of an approximation scheme proposed by Garg and Konemann in their work [7] on maximal multi-commodity flows.

Throughout the paper we assume that the capacity of the nominal flows is separated from the capacity used by the protection flows. There is a strong motivation for such assumption following the practical aspects of network operations. Having reserved capacity for nominal flows and even established connections, a network operator is able in fact to immediately restore nominal path once a failure is removed. Such capacity model is described as *no stub-release*, since it does not allow protection flows to use the nominal capacity that could be released due to failing nominal paths.

The paper is organized as follows. In Section 2. formulate the problem as a mathematical programming problem. Related pricing problem is discussed in Section 3.. Section 4. stress details of the proposed nominal path generation schemes. The paper is summarized with the discussion on numerical experiments (Section 5.) and conclusions (Section 6..

## 2. Problem formulation

Let us denote a network graph by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges, respectively. Failure network states are denoted by $\mathcal{S}$ and for each state $s \in \mathcal{S}$, set $\mathcal{E}_s$ is defined as a set of links available in this state. Each traffic demand ($d$ say) is represented by a triple $(a(d), b(d), h_d)$ meaning starting node, terminating node, and the volume of demand $d$. Given demands must be realized in any network failure state (set of failure states is denoted by $\mathcal{S}$) using nominal or backup flows. Flows of demand $d$ can use only paths from a set of candidate path pairs, denoted by $\mathcal{P}_d$. Thus, we define continuous variables $x_p$ as flows transmitted on nominal path $p$ when available and on the corresponding backup path $q$, otherwise. Nominal path availability in specific network state $s$ is defined through value of constant $\theta_{ps}$, i.e., nominal path belonging to pair $p$ is working in state $s$ if $\theta_{ps} = 1$. Path-link incidence of nominal and backup paths are described through constants $\delta_{ep}$ and $\delta_{eq}$. In according to no stub-release capacity model, the nominal capacity is separated from the protection capacity. However, cost of a unit portion of capacities both types is the same and denoted by $\xi_e$. Hence, we do use one variable $y_e$ to denote the demanded capacity of link $e$. General problem of link dimensioning assumes minimization of capacity cost expressed as $\sum_{e\in\mathcal{E}}\xi_e y_e$. Using the introduced notation we can formulate the considered problem in the L-P notation as follows.

**objective function**

$$\textit{minimize } f(\boldsymbol{x}, \boldsymbol{y}) = \textstyle\sum_{e\in\mathcal{E}}\xi_e y_e \tag{1a}$$

**constraints**

$$\textstyle\sum_{p\in\mathcal{P}_d} x_p = h_d \qquad d \in \mathcal{D} \tag{1b}$$

$$\textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}_d}(\delta_{ep} + \delta_{eq}(1-\theta_{ps}))x_p \le y_e \qquad s \in \mathcal{S},\ e \in \mathcal{E}_s. \tag{1c}$$

Note that the optimal solution of the above formulation is globally optimal if the lists of the candidate path pairs contain all possible path pairs. In other case, whether the optimal solution of the above formulation enabling limited candidate pair lists is also an optimal solution of the formulation enabling all possible candidate pair lists, depends on defined path pairs. The main idea behind using the path generation technique is to keep the lists as short as possible while required path pairs are generated when needed. Path generation makes use of properties following the duality theory of linear programming. In the next section we present the dual of (1). On the basis of this dual we are able to formulate the pricing problem and discuss its applicability.

## 3. Pricing problem

Let $\lambda_d$ and $\pi_{es}$ be the dual multipliers assigned to constraints (1b) and (1c), respectively. Using these multipliers we can formulate the dual of (1) as presented in the following.

**objective function**

$$\textit{maximize}\ g(\boldsymbol{\lambda}, \boldsymbol{\pi}) = \sum_{d\in\mathcal{D}} h_d \lambda_d \tag{2a}$$

**constraints**

$$\sum_{s\in\mathcal{S}} \pi_{es} = \xi_e \qquad e \in \mathcal{E} \tag{2b}$$

$$\lambda_d \leq \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}_s} (\delta_{ep}\pi_{es} + (1-\theta_{ps})\delta_{eq}\pi_{es}) \qquad d \in \mathcal{D},\ p \in \mathcal{P}_d. \tag{2c}$$

Observe that the expression on the right hand side of constraint (2c) can be treated as *generalized length* of path pair $p \in \mathcal{P}_d$. It consists of two components: length of the nominal path $\sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}_s} \delta_{ep}\pi_{es}$ calculated with respect to link weights $\sum_{s\in\mathcal{S}} \pi_{es} = \xi_e$ and length of backup path $\sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}_s} (1-\theta_{ps})\delta_{eq}\pi_{es}$ calculated with respect to link weights $\pi_{es}$ aggregated over those failure states for which the nominal path is not working. Assigning each path pair from candidate list of demand $d$ such generalized length, we conclude that due to maximization of $\sum_{d\in\mathcal{D}} h_d\lambda_d$, variable $\lambda_d$ is equal to the shortest *path pair length* in according to introduced generalized path pair length. Having given a path pair with generalized length smaller than actual value of $\lambda_d$, current optimal value of the objective function can be further increased through extension of the appropriate candidate list with the given path pair. This observation is in fact fundamental property exploited by the path generation technique. Let $(\boldsymbol{\lambda}^*, \boldsymbol{\pi}^*, \boldsymbol{\mu}^*)$ be the optimal solution of dual (2) obtained for limited path pair lists $\boldsymbol{\mathcal{P}}^* = \{\mathcal{P}_0^*, \mathcal{P}_1^*, \ldots, \mathcal{P}_D^*\}$. Having given a path pair $p^0$ of demand $d$ such that the value of expression $\sum_{e\in\mathcal{E}} \sum_{e\in\mathcal{E}_s} \delta_{ep}\pi_{es}^* + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}_s} (1-\theta_{ps})\delta_{eq}\pi_{es}^*$ is smaller than $\lambda_d^*$, i.e., with generalized length shorter than generalized length of any path pair from demand $d$ list, we can reformulate the dual adding new row, corresponding to path pair $p^0$. Thus, considering solution $(\boldsymbol{\lambda}^0, \boldsymbol{\pi}^*, \boldsymbol{\mu}^*)$, where $\boldsymbol{\lambda}^0$ is equal to $\boldsymbol{\lambda}^*$ on all positions except $\lambda_d^0 = \lambda_d^* - \delta$, that is equal to the generalized length of path pair $p^0$, the corresponding value of the objective function is equal to $g^0(\boldsymbol{\lambda}^0, \boldsymbol{\pi}^*) = \sum_{d\in\mathcal{D}} h_d\lambda_d^0$ that is smaller than original value of the objective function $g^*(\boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)$. Since point $(\boldsymbol{\lambda}^0, \boldsymbol{\pi}^*)$ is obviously a feasible solution, the corresponding value of the objective function is in fact a lower bound on the objective function value of the modified problem. Although, the optimal value of the objective function may be bigger, it cannot exceed value $g^*$, due to more constrained solution space. Basically, general conclusion that follows from the above considerations is that introducing path pair $p^0$ on candidate path pair list we can decrease the optimal value of the dual objective function making its value closer to the optimal value of the objective function of the primal problem with candidate list containing all path pairs. From the other hand, there is no guarantee that introduction of path pair $p^0$ with generalized length shorter than the corresponding value $\lambda_d^*$ does decrease the optimal objective function value of the dual. However, it is still acceptable to extend the candidate lists with reasonable number of non-improving path pair, if only the overall procedure allows to obtain the optimal solution of the problem enabling all possible path pairs (but without introducing them on the lists.)

Now let us proceed to the discussion on the problem of generating new path pairs. Having given link weights $\pi_{es}^*$, the pricing problem consists in determining a path pair such that

the corresponding generalized path pair length is smaller than optimal value of $\lambda_d^*$. Since, the second component (related to the backup path length) of the expression determining generalized path pair length, depends on failure states in which a nominal path is unavailable, the pricing problem is not trivial. However, it can always be formulated as a MIP problem. Let $x_e$ and $y_e$ denote the nominal and backup path incidences, respectively. Then, considering demand $d$ and dual multipliers $(\boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)$ the pricing problem can be stated as in the following.

**objective function**

$$\textit{minimize } h(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{e \in \mathcal{E}} \xi_e x_e + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}_s} \pi^*_{es} z_{es} \tag{3a}$$

**constraints**

$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = \varepsilon_{vd} \qquad v \in \mathcal{V} \tag{3b}$$

$$\sum_{e \in \delta^+(v)} y_e - \sum_{e \in \delta^-(v)} y_e = \varepsilon_{vd} \qquad v \in \mathcal{V} \tag{3c}$$

$$y_e + x_f \leq 1 \qquad s \in \mathcal{S},\ e, f \in \mathcal{E}_s \tag{3d}$$

$$z_{es} \geq y_e - x_e \qquad s \in \mathcal{S},\ e \in \mathcal{E}_s \tag{3e}$$

$$z_{es} \in [0, 1] \qquad s \in \mathcal{S},\ e \in \mathcal{E}_s \tag{3f}$$

$$x_e \in \{0, 1\} \qquad e \in \mathcal{E} \tag{3g}$$

$$y_e \in \{0, 1\} \qquad e \in \mathcal{E}. \tag{3h}$$

Formulating the objective function in the above formulation, we used equality (1b) to simplify expression determining generalized path pair length. Since the problem is formulated in the N-L notation, not used as far as now, a short description of the formulation might be desired. So, the formulation is based on flow conservation constraints for nominal (3b) and backup (3c) paths. Both path realizations must be unique (unsplittable) and disjoint in according to failure states. Thus, flows $x_e$ and $y_e$ are defined using binary variables and due to constraints (3d) only one of them may be positive for any combination of links unavailable due to one failure (especially, the paths must be link disjoint). Another formulation of this problem can be found in [1]. Formulation proposed by the authors of [1] uses two big-M constraints in the place of potentially many constraints (3d). However, the presented formulation uses much less binary variables.

Usefulness of the presented pricing problem formulation seems to be limited, because of the high complexity. Hence, we consider approximated solutions based on predefinition of the candidate nominal path sets. Having sets of the nominal paths defined, we can reduce the pricing problem to simple enumeration of the nominal paths from the lists and determination of the shortest backup path for each nominal path with respect to aggregated over states $\pi_{es}$. If the value of the generalized path pair length of such a path pair is smaller than value of the corresponding $\lambda^*$, then the path pair should be introduced on an appropriate path pair list. As numerical experiments, presented in Section 5., shows such approach seems to be a reasonable approximation, often allowing to obtain the globally optimal solution of

the considered problem. Still, the effectiveness of the proposed simplified problem heavily depends on the definition of the nominal path sets. That is why in the sequel we focus on this problem and propose two methods of solving it.

## 4. Nominal path precomputation

Due to assumed simplification of the pricing problem, an important aspect of solving the considered dimensioning problem is to determine an initial set of *good* nominal candidate paths. Having routing lists that contain *good* paths (sufficient for resolving the problem in hand) the resolution process would be simplified a lot. It seems that a reasonable approach for arriving at lists of paths which have a good chance to be sufficient for solving the problem can be based two approaches: k-shortest path algorithm and approximation scheme presented in [7]. In this section we stress these two propositions of predefinition of the nominal path sets. Basically, both methods are iterative procedures built on the strength of a shortest path algorithm. Hence, they are expected to be very efficient and scalable.

### 4.1. K-shortest path algorithm

The first proposed method of populating the nominal path lists is the application of a k-shortest path algorithm with link weights equal to one. We believe that in the dimensioning problem, *good* paths should be rather short with respect to the hop count measure. Hence, using weights equal to one, we can construct the lists of potentially *good* nominal paths that are likely to be used in the globally optimal solution. This conviction follows observation that a situation when a nominal path must use relatively long path consuming a lot of link capacity is quite rare. Since, the k-shortest path algorithms are satisfactorily recognized in the literature we skip characterization of the algorithms solving this problem. For more details see [13], [9], [6], and [5].

### 4.2. Approximation scheme

The second proposition of the method populating the nominal path lists is application of Fully-Polynomial Approximation Scheme (FPTAS) approach proposed by Garg and Konemann in their work [7]. The idea has been developed for a family of approximate algorithms for solving a number of multi-commodity flow problems which are special cases of the linear packing problems. These are primal-dual algorithms which exploit a common general idea of bridging the gap between primal and dual solutions by repeatedly allocating flows to the shortest paths and modifying costs (weights) of links. Such algorithms can be used, for example, to solve the problems of maximizing the total multi-commodity flow, the common flow, the cost-bounded common flow, etc. The general scheme of all these algorithms is the same and they are devised in a similar way.

In general, the considered problem is a capacitated multi-commodity flow problem in the link-path formulation. It involves a vector of path flow variables $\boldsymbol{x} = (x_p : d \in \mathcal{D},\ p \in \mathcal{P}_d)$,

an objective function $f(\boldsymbol{x})$ to be maximized, and a set of link capacity constraints:

$$\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}_d}\delta_{ep}h_d x_p \le c_e \qquad e\in\mathcal{E}. \tag{4}$$

The dual problem is a kind of link cost assignment problem. It involves dual variables $\pi_e$ corresponding to the capacity constraints (4) of the primal problem. Let $\boldsymbol{\pi} = (\pi_e \ge 0 : e \in \mathcal{E})$ be the vector of these dual variables. The objective of the dual problem is to minimize the dual objective function $W(\boldsymbol{\pi}) = \sum_{e\in\mathcal{E}} c_e\pi_e$, where $c_e$ is the capacity of link $e$. This is subject to a constraint that for each demand the length of its shortest path is lower bounded by a positive value which is related to the coefficients of the primal objective function.

Let $\lambda_d(\boldsymbol{\pi})$ denote the length of the shortest path with respect to weights $\boldsymbol{\pi}$ on the path list of demand $d$, i.e., $\lambda_d(\boldsymbol{\pi}) = \min_{p\in\mathcal{P}_d}\{\sum_{e\in\mathcal{E}}\delta_{ep}\pi_e\}$. The dual problem is then equivalent to finding a variable assignment $\boldsymbol{\pi} \ge \mathbf{0}$ such that $\frac{W(\boldsymbol{\pi})}{w(\boldsymbol{\pi})}$ is minimized, where $w(\boldsymbol{\pi}) = \sum_{d\in\mathcal{D}}\lambda_d(\boldsymbol{\pi})$.

The algorithm is iterative. At the start of the algorithm a uniform weight $\Delta > 0$ is assigned to all links. Then, at each iteration, flows are assigned to shortest paths according to some procedure which depends on the exact problem formulation, in particular on the form of the objective function. Every time flow $f > 0$ is assigned to a link of capacity $c$, the weight of that link is multiplied by $1 + \varepsilon f/c$, where $\varepsilon$ is a carefully selected positive constant. The iterations are performed as long as the weights of all links are less than 1.

Let $\boldsymbol{x}^i$ and $\boldsymbol{\pi}^i$ denote, respectively, the values of the primal path flow variables and the dual link weight variables at the start of iteration $i$. Let $f_e^i \le c_e$ be the amount of flow assigned to link $e$ in iteration $i$. Then $\pi_e^1 = \Delta$ and $\pi_e^k = \pi_e^{k-1}(1 + \varepsilon f_e^{k-1}/c_e) = \Delta\prod_{i<k}(1 + \varepsilon f_e^i/c_e)$ for $k > 1$.

At the start of the algorithm the value of the dual objective function equals $W(\boldsymbol{\pi}^1) = \sum_{e\in\mathcal{E}} c_e\Delta$, and the value of $W$ at the start of iteration $i > 1$ can be expressed with the following recursive formula:

$$W(\boldsymbol{\pi}^i) = \sum_{e\in\mathcal{E}} c_e\pi_e^i = \sum_{e\in\mathcal{E}} c_e\pi_e^{i-1}(1 + \varepsilon f_e^{i-1}/c_e) = W(\boldsymbol{\pi}^{i-1}) + \varepsilon\sum_{e\in\mathcal{E}}\pi_e^{i-1}f_e^{i-1}. \tag{5}$$

If the recurrence is resolved one gets:

$$W(\boldsymbol{\pi}^i) = W(\boldsymbol{\pi}^1) + \varepsilon\sum_{k<i}\sum_{e\in\mathcal{E}}\pi_e^k f_e^k. \tag{6}$$

The sum $\sum_{e\in\mathcal{E}}\pi_e^k f_e^k$ expresses the total dual cost of the flows allocated in iteration $k$. If in each iteration the flows are assigned to the shortest paths of the demands, this sum can be rewritten as $\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}_d}\lambda_d(\boldsymbol{\pi}^k)(x_p^{k+1} - x_p^k)$. Since the dual weights are non-decreasing, we have that $\lambda_d(\boldsymbol{\pi}^i) \le \lambda_d(\boldsymbol{\pi}^j)$ for $i \le j$, and:

$$\sum_{k<i}\sum_{e\in\mathcal{E}}\pi_e^k f_e^k = \sum_{k<i}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}_d}\lambda_d(\boldsymbol{\pi}^k)(x_p^{k+1}-x_p^k) \le \sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}_d}\lambda_d(\boldsymbol{\pi}^{i-1})x_p^i. \tag{7}$$

Thus, finally, one gets:

$$W(\boldsymbol{\pi}^i) \le W(\boldsymbol{\pi}^1) + \varepsilon\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}_d}\lambda_d(\boldsymbol{\pi}^{i-1})x_p^i. \tag{8}$$

The link weight modification rule means that in practice the weight of a link can be multiplied by at most $1+\varepsilon$ in a single iteration. Thus, when the algorithm stops all weights of links are less than $1+\varepsilon$, because before the last iteration they all are less than 1. If in each iteration at least one link is assigned the flow equal to the link capacity (its weight is thus multiplied by $1+\varepsilon$) the total number of iterations is at most $E\log_{1+\varepsilon}\frac{1+\varepsilon}{\Delta}$.

At the end of the algorithm the total amount of flow $\sum_i f_e^i$ on link $e$ is not greater than $c_e\log_{1+\varepsilon}\frac{1+\varepsilon}{\Delta}$, because, due to the fact that $f_e^i/c_e \le 1$, the following relations hold:

$$\Delta(1+\varepsilon)^{\sum_i f_e^i/c_e} = \Delta\prod_i(1+\varepsilon)^{f_e^i/c_e} \le \Delta\prod_i(1+\varepsilon f_e^i/c_e) \le 1+\varepsilon. \tag{9}$$

It follows that if all flows are scaled down by $\log_{1+\varepsilon}\frac{1+\varepsilon}{\Delta}$ one obtains a feasible multicommodity flow.

For each demand the paths that were used by the algorithm can be ordered with respect to the total amount of flow they were assigned. This ordering can be viewed as a ranking of usefulness of the generated paths. For each demand a set of paths with the highest ranking can then be selected for the future use as an initial set of the candidate paths.

## 5. Numerical experiments

In our computational investigations we perform a series of performance tests of discussed methods. For this purpose we selected a set of 3 interesting network instances, ranging in size from 5 to 20 nodes. The network topologies as well as traffic demands and capacity costs were generated randomly. In our experiments we consider all single link failures. Bundle link failures are left for now because experiments for such failure scenarios demand careful preparation of the input network data. In fact to obtain realistic failure model it is desired to consider at least two-layer network model.

To solve the MIP pricing problem we had programmed it using the Callable Library API 1.2 and then treated it with CPLEX 10.1. Thus, the MIP problem was resolved by means of the CPLEX's built-in B&C solver and the original problem of resilient routing optimization was resolved by means of our own path generation framework combined with CPLEX's LP solver. Generation of the nominal paths were done once using a k-shortest path algorithm based on the Dijkstra shortest path algorithm and second using the approximation scheme

described in Section 4.2.. Since the assumed scenario demands that only backup paths are generated in the path generation framework, the path pair generation reduces to iterative invocation of a shortest path algorithm. In our case, the shortest paths are calculated using the Floyd shortest path algorithm.

First performed experiment concern determination of satisfying number of the nominal paths. In this experiment we ran the path generation framework procedure in a few scenarios assuming the number of the nominal paths limited from one to ten. Experiment is performed using three example networks: 5-node network donted by n5, 10-node network denoted by n10, and 20-node network denoted by n20. The results of the experiment are presented in Table 1.

| Network | # nominal paths | objective function value | # generated backup paths |
|---|---|---|---|
| n5 | 1 | 567.2 | 10 |
| n5 | 2 | 567.2 | 10 |
| n5 | 5 | 567.2 | 10 |
| n5 | 10 | 567.2 | 10 |
| n10 | 1 | 4184.25 | 13 |
| n10 | 2 | 4180 | 14 |
| n10 | 5 | 4180 | 14 |
| n10 | 10 | 4180 | 14 |
| n20 | 1 | 16523.5 | 30 |
| n20 | 2 | 16520.1 | 40 |
| n20 | 5 | 16516.9 | 31 |
| n20 | 10 | 16516.9 | 34 |

Table 1. Objective function value in relation to the number of nominal paths

Observe that limiting the number of the nominal paths to the half of the number of nodes, we are still able to obtain satisfying values of the objective function. Assuming in the sequel that the number of nominal paths is limited to the half of the number of nodes, we performed the second numerical experiment. In this test, we compare three approaches: exact pricing (column "MIP pricing"), nominal paths predefinition on the basis of k-shortest path solution (column "K-shortest"), and nominal paths predefinition on the basis of approximation scheme (column "Approx"). We compare in fact two quantities: computation time and value of objective function. Results of this experiment are gathered in Table 2. To name the network instances we use the previously introduced notation.

| | MIP pricing | | K-shortest | | Approx | |
|---|---|---|---|---|---|---|
| Network | Time [sec.] | Obj. value | Time [sec.] | Obj. value | Time [sec.] | Obj. value |
| n5 | 1 | 567.2 | 1 | 567.2 | 1 | 567.2 |
| n10 | 29 | 4180 | 1 | 4180 | 1 | 4180 |
| n20 | 2580 | 16516.9 | 739 | 16516.9 | 690 | 16516.9 |

Table 2. Comparison of two approximated path pair generation solutions with the exact method

As Table 2 shows, in all considered cases the obtained solutions are globally optimal (proved by the exact pricing). Hence, the general conclusion is that it is a reasonable assumption to

generate the nominal path sets in the preprocessing phase (i.e., using one of the proposed scheme) and use exact generation of backup paths coupled with given nominal paths. We believe that the chance to obtain the globally optimal solution using such a method is quite high. It happens so because despite the nominal paths may be not the best but through careful generation of appropriate backups we are still able to obtain the cheapest flow distribution pattern.

## 6. Conclusions

Paper discuss the problem of link dimensioning in the resilient networks with protection on single backup paths. The problem is formulated in the link-path notation and a path pair generation method supporting this formulation are proposed. Since, the exact pricing problem may be inefficient in some network applications, we proposed two variants of nominal path precomputation combined with path generation of backup paths for these nominal paths.

As numerical experiments, stressed in Section 5., shown solving the problem of link dimensioning in resilient networks with protection on single backup paths can be done quite efficiently on the basis of approach proposed in this paper. Numerical results obtained through application of the simplified pricing problem are close to the optimal solution. Hence, the proposed methods seems reasonable alternative for exact pricing problem demanding solving of a MIP problem.

In the future work we would like to define a set of two-layer networks, so to test efficiency of the presented methods for the case of multiple link failures. In the circle of our interests remains also application of the exact algorithm solving the pricing problem on the basis of label setting algorithm, proposed by Stidsen et al. in [1].

## References

[1] Optimal routing with single backup path protection. In *International Network Optimization Conference INOC2007, Spa, Belgium*, 2007.

[2] A. Bashllari, D. Nace, E. Rourdin, and O. Klopfenstein. The MMF rerouting computation problem. In *International Network Optimization Conference INOC2007, Spa, Belgium*, 2007.

[3] R. Bhandari. *Survivable Networks – Algorithms for Diverse Routing*. Kluwer, 1999.

[4] D. Coudert, P. Datta, S. Perennes, H. Rivano, and M.-E. Voge. Complexity and approximability issues of shared risk resource group. Technical report, 2006. Technical report 5859, INRIA.

[5] S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17:395—412, 1999.

[6] D. Eppstein. Finding the k shortest paths. In *$35^{th}$ IEEE Symposium on Foundations of Computer Science*, pages 154–165, 1994.

[7] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[8] J.Q. Hu. Diverse routing in optical mesh networks. *IEEE Trans. Com.*, 51(3):489–494, 2003.

[9] V. M. Jiménez and A. Marzal. Computing the k shortest paths: A new algorithm and an experimental comparison. In *Proceedings of 3$^{rd}$ Annual Workshop on Algorithmic Engineering, London*, 1999.

[10] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[11] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4:125–145, 1974.

[12] Y. Yang and J.Wang. Routing permutations with link-disjoint and node-disjoint paths in a class of self-routable networks. In *IEEE International Conference on Parallel Processing (ICPP'02), Vancouver*, pages 154–165, 2002.

[13] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 259 - 268

# Comparison of centralized and decentralized preemption in MPLS networks

SYLWESTER KACZMAREK [a] KRZYSZTOF NOWAK [b]

[a] Faculty ETI, Department STI
Gdańsk University of Technology
*Sylwester.Kaczmarek@eti.pg.gda.pl*

[b] Nokia Siemens Networks Sp. z o.o.
*krzysztof.nowak@nsn.com*

**Abstract:** Preemption is one of the crucial parts of the traffic engineering in MPLS networks. It enables allocation of high-priority paths even if the bandwidth on the preferred route is exhausted. This is achieved by removing previously allocated low-priority traffic, so as enough free bandwidth becomes available. The preemption can be performed either as a centralized or a decentralized process. In this article we discuss the differences of both approaches. We present a series of simulation results to show the performance of selected methods for different network topologies.

**Keywords:** MPLS, preemption, traffic engineering, performance, simulation

## 1. Introduction

In the late 90's the Internet Engineering Task Force (IETF) formulated requirements for the traffic engineering in MPLS networks. The related document [1] identifies the most important methods necessary to implement efficient network management, including preemption. Being only a general guide, the document neither specifies how to implement it nor proposes any specific preemption method. However, since its publication, a number of papers have become available on different preemption methods. One of them has recently been accepted as an informational RFC [2].

The majority of authors proposes decentralized preemption methods, as they seem to suit better to decentralized Internet architecture and are easier to implement. There are also several centralized methods, developed to take advantage of broader view on the paths routed in the network. However, to the best knowledge of the authors, no comparison of these two approaches is available. Yet such information might be a valuable resource for anyone who plans to implement preemption in a specific network. The purpose of this paper is to give the reader a general view on the subject.

The structure of the paper is as follows. First we explain, how preemption works and what the differences between centralized and decentralized approach are. We describe

their potential advantages and disadvantages. The third section is devoted to the simulation scenarios and metrics we used to evaluate the methods. We also included a short description of the selected methods. In the fourth section we present and discuss the results. Finally in conclusions we try to formulate basic rules for selecting preemption approach and directions for future works in this area.

## 2. Preemption basics

Preemption can be seen as an extension to a bandwidth-aware routing protocol, like the Constraint Shortest Path First (CSPF). Without preemption, when allocating a path of priority $p$ with bandwidth $b$, only the links with free bandwidth of at least $b$ are taken into consideration. With preemption, the free bandwidth on every link is increased by the bandwidth used by the paths of lower priority, i.e. numerically greater than $p$.

Generally a preemption method begins with checking, on which links there is not enough free bandwidth and calculating how much of it is missing, to accommodate a new path. Than the preemption algorithm selects a set of candidates, i.e. paths which are to be preempted, with the priority lower than the new path. This process varies depending on the specific preemption method. The selected candidates are removed so as the route can have enough free bandwidth and the new path can be allocated. Finally the removed candidates are allocated again on alternative routes.

In a centralized preemption procedure one dedicated device selects the candidates along the route. It uses the full information about the available paths including their location to choose the set of candidates with the lowest cost. This is not possible with a decentralized method, where the routers have information about the local paths only. In this case the candidates are selected using a goal function which minimizes the cost locally. As soon as the selection is made, the candidates are removed from the network. In case of a decentralized method the procedure is repeated on the next router, until there is enough free bandwidth along the whole route.

Both attempts have advantages and disadvantages. If we consider the necessary effort to implement a method, it should usually be lower for a decentralized solution, as there is no need to store in one of the routers the whole path information. When we think about the performance, a centralized method might select the best set of candidates. However, to get better results, centralized methods need at least two links along the route, where the preemption is needed. In other words, it will not perform better if preemption will rarely be needed on more than one link along the route. Better results can possibly be achieved in complex networks, especially under high load. We will discuss later, in which cases centralized methods perform better.

There is a number of metrics used to evaluate preemption methods, but two of them are most commonly used. The first one is the relocation count, defined as the number of paths which are selected for the candidates and removed. In real networks every preemption results in some traffic loss and therefore should be avoided when not necessary.

The second metric is the preempted bandwidth, calculated simply as the sum of bandwidths of the preempted paths. A modified metric is called the preempted network bandwidth, for which the bandwidth of every path is multiplied by its hop count. In the context of preempted bandwidth one can also say about the bandwidth wastage. This is the difference between the preempted network bandwidth and the bandwidth which is really needed. In the ideal situation there is no wastage – that is, the preempted bandwidth is equal to the needed bandwidth. However, in reality the bandwidth wastage is unavoidable, because usually there are no paths, which would free exactly the needed bandwidth. What can be done, it to minimize the unnecessarily preempted bandwidth.

## 3. Simulation scenario

The problem of selecting the lowest cost set of candidates is known to be *NP*. That makes the optimal solution infeasible to be performed online. Instead, heuristic preemption methods have been developed. We chose two of them for comparison: decentralized RFC method [2] and our own heuristic centralized method [3]. In the rest of the paper we use symbols "DEC" and "CEN", respectively. These two methods were selected for they both are clearly defined and adjustable. For both methods we used two different goal functions: to minimize relocation count and to minimize the preempted bandwidth. For these variants we use symbols "REL" and "BW", respectively.

In the decentralized (DEC) method, every path $l$ is assigned a cost $H(l)$ and the available paths are sorted in increasing order. Paths with the same value of $H(l)$ are ordered by the bandwidth $b(l)$, in increasing order. The definition of $H(l)$ is as follows:

$$H(l) = \alpha\, y(l) + \beta \frac{1}{b(l)} + \chi \left(b(l) - r\right)^2 + \theta\, b(l), \qquad (1)$$

where $y(l)=8-p(l)$ is the priority function, $p(l)$ is the priority of the path, $r$ is the needed bandwidth, i.e. the difference between the requested bandwidth and free bandwidth on the corresponding link. The coefficients $\alpha$, $\beta$, $\chi$, and $\theta$ are input values to adjust the goal function. In our case the REL variant corresponds to $\alpha=\chi=\theta=0$, and $\beta=1$, whereas the BW variant corresponds to $\alpha=\beta=\chi=0$, and $\theta=1$.

The centralized method (CEN) performs in the following way. For every path which follows any of the links where preemption is needed, three steps are done. First, if the path can be preempted, it is added to the list of candidates without any additional check. Second, the bandwidth of the path is added to the available bandwidth for every links it uses. Finally, the list of candidates is being checked against unnecessary paths. This is achieved by temporarily removing candidates one by one from the list and checking of how much has available bandwidth dropped. If there still remains enough bandwidth, the path is excluded form the list. Otherwise, it remains as the candidate and the next path is checked. The details about the method can be found in [3].

The CEN method is adjustable by setting the sorting method, what influences the order, in which the paths are removed from the list of candidates. For the REL variant, we sort paths by the bandwidth, in increasing order, so as the smaller paths are excluded first and lower number of paths remains on the list. For the BW variant, we perform the

sorting in decreasing order, what makes the bigger paths being excluded first and smaller paths remain as the candidates. Note that the sorting order is equivalent to the *H(l)* adjustment which we use for the DEC method.

We compared the methods using the *MPLSsim* simulation program, which we developed entirely to simulate MPLS networks. For the purpose of preemption evaluation we used connection level simulations.

To check the performance of the methods for different topologies, we chose several networks:

1. Large random network. This is a random topology of 100 nodes, including 20 edge routers, created using the locality [4] variant of the Waxman topology generation method. The topology generator was programmed to create bidirectional connections with the following parameters: near probability of 0.3, far probability of 0 (zero) and distance multiplier of 0.2. The resulting network shown in Fig. 1 has the average connectivity index of 5.24 and the average hop count of 3.19 (calculated using the Shortest Path First algorithm).
2. Ring topology. We chose a 12-nodes bidirectional ring as shown in Fig. 2.
3. Mesh topology (unidirectional). The regular network consists of 24 nodes and is shown in Fig. 3.

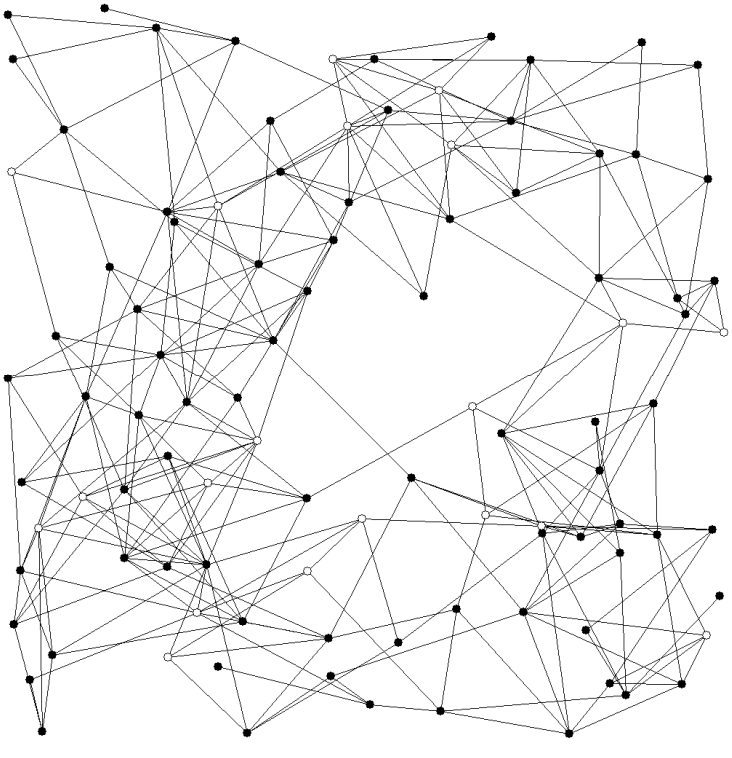


Fig. 1. Random topology used for simulation. White-filled nodes are the edge routers.

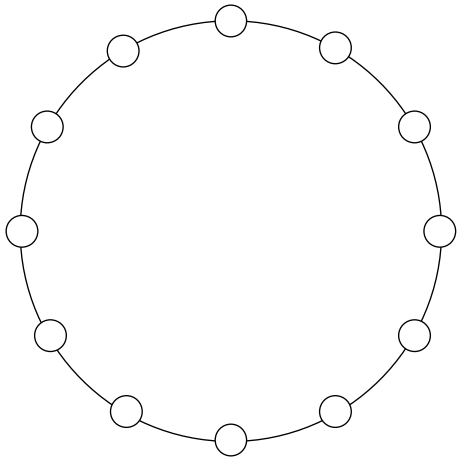


Fig. 2. Ring topology used for simulation.

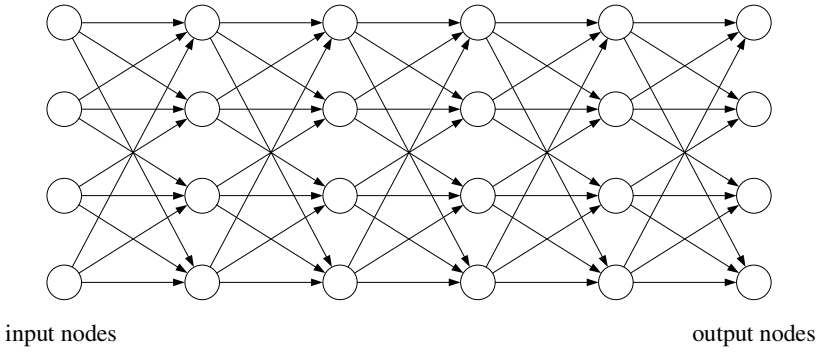



Fig. 3. Mesh topology used for simulation.

Path allocation requests were generated at random time points with exponential distribution with an average value calculated from the adjustable parameter of "sources per second". The egress and the ingress node were chosen randomly from the edge nodes. The request was assigned in the round-robin manner to one of three traffic classes, numbered 1, 2 and 3 with corresponding priority 1, 2 and 3. The lower number means higher priority, e.g. a path of priority 1 can preempt paths of priorities 2 and 3, but it can not be preempted by any other path. The bandwidth of the path was chosen randomly with exponential distribution of a given mean value. The parameters used in simulations are shown in Tab. 1.

We compared the methods using the following metrics.

1. Number $r_{al}$ of preempted paths per allocation. This is the average number of paths which are removed as a result of the new allocation, not including the reallocations,

i.e. paths allocated as the results of previous preemptions. The lower value we get here, the better the method is, because it keeps more existing paths from being removed. This is a modified relocation count metric, so we use the terms interchangeably in the rest of the paper.

2. Sum $s_n$ of network bandwidth of preempted paths. The network bandwidth $b_n$ of a path is the bandwidth reserved for the path multiplied by the length (hop count) of the path. That is:

$$s_n = \sum_{k=1..K} c_k l_k \ , \tag{2}$$

where $K$ is the number of paths deleted during a single preemption, $c_k$ is the bandwidth allocated to $k$-th preempted path and $l_k$ is the length (hop count) of the path. The lower value of $s_n$ means less bandwidth removed to allocate new paths, what results in more economical bandwidth management.

3. Combined quality metric $q$ using both factors: the number of preempted paths and the preempted network bandwidth. The metric is defined as:

$$q = \frac{d_n}{r \cdot b_n}, \tag{3}$$

where $d_n$ is the bandwidth deficit along the route, calculated as the sum of differences between the bandwidth of new path and the free bandwidth on every link along the selected route; $r$ is the relocation count, and $b_n$ is the network bandwidth removed. The metric is defined in the way that greater value of $q$ means better method in the sense that small number of preempted paths is combined with low bandwidth wastage.

| Topology | Avg. path bandwidth | Sources per second |
|---|---|---|
| Random | 15 Mb/s | 100 |
| Ring | 4 Mb/s | 100 |
| Mesh | 2 Mb/s | 100 |

Tab. 1. Traffic generator parameters used in simulations

## 4. Results

In figures 4, 5 and 6 we present the results of $r_{al}$, $s_n$, and $q$, respectively. First we show results of the number of preemptions per allocation. The lower values here mean better results, i.e. fewer paths are preempted to get enough free bandwidth. From that perspective, we do not care about the potential bandwidth wastage.

The results presented in Fig. 4 may seem surprising, as both DEC and CEN methods perform comparably well, provided that they are adjusted to the preemptions priority (REL variants). In fact the differences are within the period of confidence, whereas we could expect the centralized method to perform better. The explanation can easily be

found if we compare the traffic conditions with gains of centralized and decentralized methods discussed earlier. In the simulation scenario we selected, the mean number of links, on which the preemption was necessary, was only about 1.31. This limited the potential strength of the DEC method and gave the reason for similar results of the methods CEN and DEC.

One more thing which we should point out to is the result of the DEC:BW method, where the relocation count value is about twice as high as the value of the other methods. As long as we remember that this method is optimized for minimizing the bandwidth wastage only, this may not be a problem. However, in comparison to other methods the relocation count might be too high in this case. In contrast, the relocation count for the method CEN:BW is not much higher than the results of both REL methods. This suggests, that it succeeds in offering the best trade-off between the number of preemptions and the preempted bandwidth.

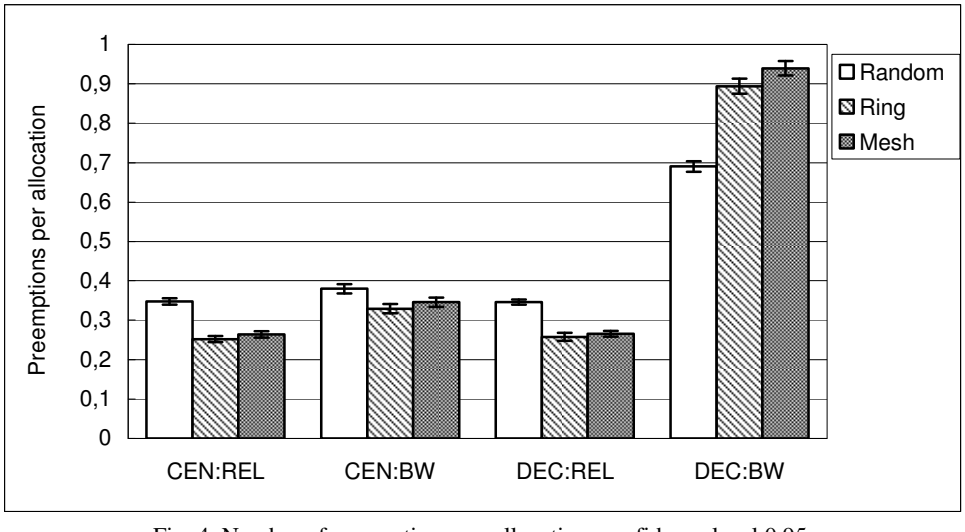



Fig. 4. Number of preemptions per allocation; confidence level 0.95.

The sum of preempted network bandwidth by means of different methods is shown in Fig. 5. For the sake of clearness we divided them into two separate graphs with different ranges of Y-axis. The lower value here implies better result, because less preempted bandwidth means less traffic to be disrupted. Moreover, minimizing the preempted bandwidth usually keeps bigger paths from being preempted. That in turn has another advantage, as reallocations of smaller paths are more likely to succeed, as less free bandwidth is needed on alternative routes and the paths can also be distributed among several paths.

The results show that the CEN:BW method performs better than DEC:BW. The difference is evident, yet not very dramatic, with gain of about 10% for each topology. We can easily notice that for both methods there is strong influence of the selected variant

on the preempted bandwidth volume. The methods optimized for minimizing the preempted bandwidth perform much better, reducing the value of about 25%.

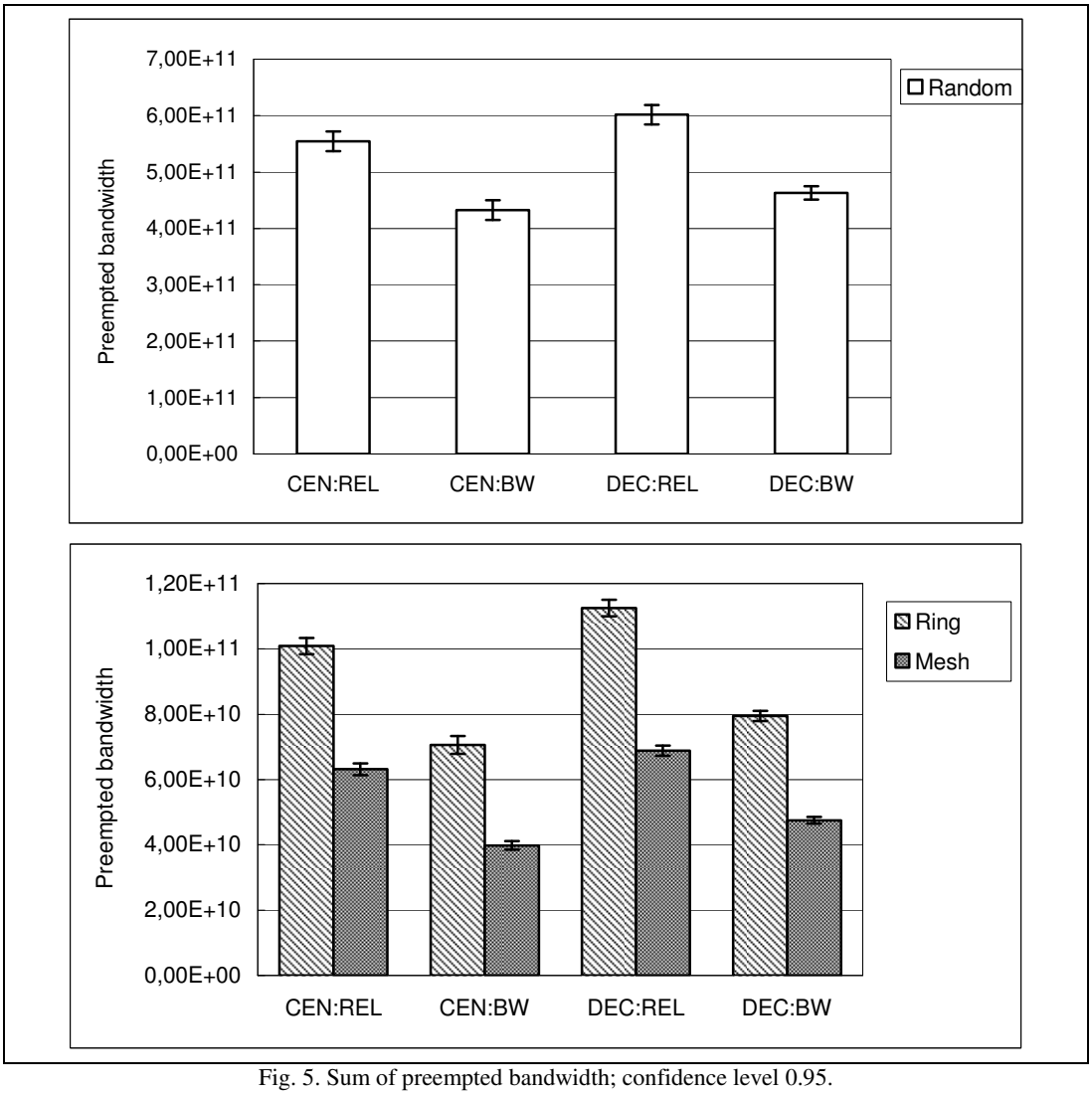



Fig. 5. Sum of preempted bandwidth; confidence level 0.95.

We get quite interesting results for our combined quality metric, as shown in Fig. 6. Opposite to the previous metrics, a greater value means better performance. The goal of this measurement is to check how universal the method is, i.e. if it keeps both the number of preemptions and the preempted network bandwidth on a low level. Here the CEN:BW method is the unquestionable winner. This is because it performs best in minimizing preempted bandwidth while keeping the number of preempted paths not much greater than the methods optimized for that. The REL variants of the methods have similar results

to each other, but the centralized method performs slightly better. The DEC:BW method is the worst here, due to the large number of preempted paths.

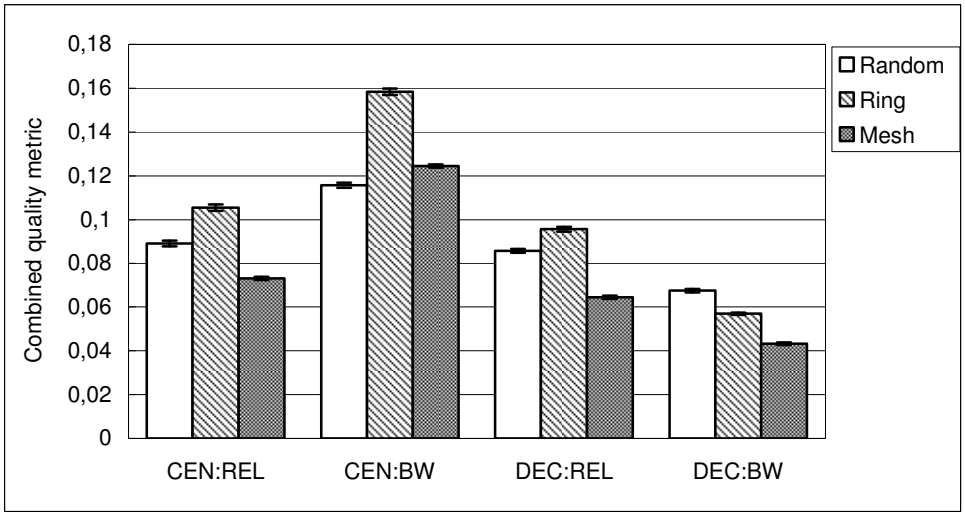



Fig. 6. Combined quality metric; confidence level 0.95.

We noted before, that the potential difference in performance between centralized and decentralized methods depend on the number of links, where the preemption is needed. If there is only one such link then the centralized method will be reduced to quasi-local method and will not show better performance. More precisely, there will be a possible difference caused by different heuristics the methods use.

To check the traffic conditions of the simulations, we measured the number of links on which there was not enough bandwidth, that is – on which the preemption was requested. Indeed, the results included in Tab. 2 show that in most cases preemption was requested on one link only. This phenomenon makes a restriction for any centralized method, lowering the potential gain in preemption efficiency. However, like we shown before, in some cases the results show considerably better results achieved with centralized methods.

| Topology | Route length | Links with preemption |
|---|---|---|
| Random | 3.69 | 1.31 |
| Ring | 3.38 | 1.30 |
| Mesh | 5.00 | 1.17 |

Tab. 2. Average route length and number of links where preemption was requested.

## 5. Conclusions

In this paper we present simulation results of centralized and decentralized preemption method. We show that the preempted bandwidth and the relocation count can vary greatly

depending on the method used for preemption. The results show that the biggest advantage of the centralized method in the CEN:BW variant is its capability to achieve the best trade-off between the relocation count and the preempted bandwidth. Also preempted bandwidth is significantly lower for the centralized method. However, both methods achieve similar results of the relocation count. We can end up with the conclusion that the centralized method will perform better when adjusting to minimize the preempted bandwidth, especially when the relocation count matters as well. If the relocation count is the only important quality factor, then both methods are comparable.

The results show that the behavior of the methods is similar for every network topology we simulated. This is an important message, because it gives us the basis for the assumption, that the conclusions hold true for many other network topologies as well.

We are going to continue the research in the area of preemption. Currently we consider implementing other known algorithms and make a broader performance comparison. There are also many factors which influence the preemption and still need to be analyzed, including unequal traffic distribution between priorities.

## References

[1] Awduche, D., et al.: *Requirements for Traffic Engineering Over MPLS*, RFC 2702, September 1999.

[2] de Oliveira, J., Ed.: *Label Switched Path (LSP) Preemption Policies for MPLS Traffic Engineering*, RFC 4829, April 2007.

[3] Kaczmarek, S., Nowak, K.: *A New Heuristic Algorithm for Effective Preemption in MPLS Networks*, 2006 Workshop on High Performance Switching and Routing, Poznań 2006, pp. 337-342.

[4] Zegura, E.W., Calvert, K.L., Donahoo, M.J.: *A Quantitative Comparison of Graph-based Models for Internet Topology*, IEEE/ACM Transactions on Networking, Vol. 3, Issue 6, Dec. 1997, pp. 770-783.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 269- 280

# Application of uncertain variables to rate allocation in the computer networks with imprecise parameters

DARIUSZ GĄSIOR [a]

[a] Institute of Information Science and Engineering
Wrocław University of Technology
Wyb. Wyspianskiego 27, Wrocław, Poland
*dariusz.gasior@pwr.wroc.pl*

**Abstract:** The paper is concerned with a problem of rate allocation in the computer network in the case when some of parameters are unknown or their values are imprecise. The network utility maximization concept is used to treat the computer network as an input-output decision making plant and formulate rate allocation problem as an optimization problem. It is assumed that an expert can describe possible values of unknown network parameters. Then formalism of uncertain variables is applied and the knowledge of an expert is modelled with certainty distributions. For such assumption three decision-making problem formulations are given and the solution algorithms are elaborated.

**Keywords:** rate allocation, uncertain variables, uncertain system

## 1. Introduction

Modern rate allocation algorithms are based on the network utility maximization concept which was introduced in [1]. Such algorithms were elaborated and studied for different computer network infrastructures – e.g. as well for wiring [2] as for wireless computer networks [3]. Two types of rate allocation algorithms can be indicated: online and offline. Online algorithms are implemented in distributed way and their goal is to react in a real-time manner for dynamically changing transmission demands and with partial (local) information about network (topology, state, etc.) available. Offline algorithms are considered centralized for fixed transmission demands (e.g. expected or mean) and for static network parameters. These algorithms are used for network planning, modeling, analyzing and as reference points for online algorithms [4]. In this paper, an offline rate allocation algorithms with QoS requirements and in the presence of unknown parameters are elaborated.

## 2. Network model

In this paper the computer networks is treated as a set of direct links between neighbour network nodes. The path is a sequence of links used for a transmission. It is assumed that there is only one path available for each transmission demand. Let us introduce the following notation:

$l \in \mathcal{L}$ – index of link,

$r \in \mathcal{R}$ – index of transmission demand,

$\mathcal{P} \subseteq \mathcal{R}$ – a set of admitted transmission demands,

$\mathcal{L}_r$ – a subset of links used by demand $r$ for the transmission,

$\mathcal{R}_l$ – a subset of transmission demands using link $l$,

$u$ – vector of data rates for all transmission demands in the network.

Every link is characterized by its bandwidth $U_l$, which expresses maximal total amount of data that can be transmitted through the link $l$ in the unit time and there is no congestion. Let $u_{\min}$ and $u_{\max}$ be respectively a vector of minimal and a vector of maximal data rates that must be allocated for transmission demands due to Quality of Service (QoS) requirements. In the network utility maximization framework it is assumed that users' preferences related to each transmission demand are described with utility functions. We assume the utility for demand *r* as follows:

$$y_r = \begin{cases} f_r(u_r; a) & \text{for} \quad u_r \geq u_{r,\min} \\ g_r(u_{r,\min}; a) & \text{for} \quad u_r < u_{r,\min} \end{cases}$$

where $f_r(u_r)$ is increasing, strictly concave and continuously differentiable function of $u_r$ over the range $u_r \geq 0$, $g_r(u_{r,\min})$ is a penalty function for not admitting transmission demand and $a$ is a vector of parameters. The total utility $y = Q(u, \mathcal{P}; u_{\min}, a)$ for the whole network is composed of utility functions for all demands and can be given e.g. in the form of weightened sum of utility functions for all demands.

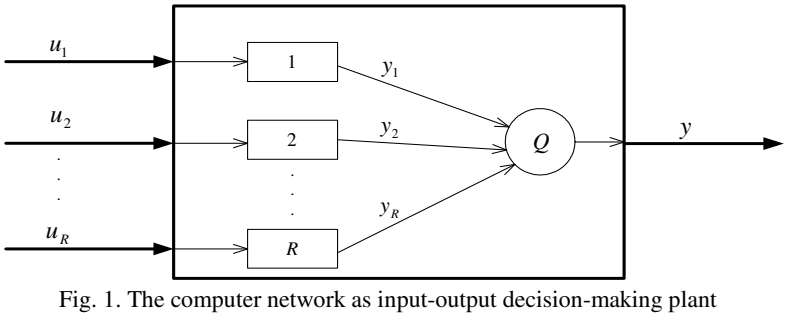



Fig. 1. The computer network as input-output decision-making plant

For such data we can treat the computer network as an input-output decision-making plant as it is shown in Fig. 1 and the decision-making problem **P1** can be formulated as an optimization problem (similarly to [5]):

**Given:** $\mathcal{R}$, $\mathcal{L}$, $\mathcal{L}_r$ for $r \in \mathcal{R}$, $\mathcal{R}_l$, $U_l$ for $l \in \mathcal{L}$, $u_{\min}$, $u_{\max}$, $a$, $Q(u, \mathcal{P}; u_{\min}, a)$

**Find:**

$$(\mathcal{P}^*, u^*) = \arg \max_{(\mathcal{P},u) \in D} Q(u, \mathcal{P}; u_{\min}, a)$$

where

$$D = \{(\mathcal{P}, u) : (\forall_{l \in \mathcal{L}} \sum_{r \in \mathcal{R}_l \cap \mathcal{P}} u_r \le U_l) \wedge (\forall_{r \in \mathcal{R} \setminus \mathcal{P}} u_r = 0) \wedge (\forall_{r \in \mathcal{P}} u_{r,\min} \le u_r \le u_{r,\max})\}.$$

It can be easily noticed that this problem can be treated as a mixed discrete–continuous optimization problem and another formulation (**P2**), can be given:

**Given:** $\mathcal{R}$, $\mathcal{L}$, $\mathcal{L}_r$ for $r \in \mathcal{R}$, $\mathcal{R}_l$, $U_l$ for $l \in \mathcal{L}$, $u_{\min}$, $u_{\max}$, $a$, $Q(u, \mathcal{P}; u_{\min}, a)$

**Find:**

$$(\mathcal{P}^*, u^*) \text{ such that } u^* = u^*(\mathcal{P}^*),$$

where:

$$\mathcal{P}^* = \arg \max_{\mathcal{P} \in 2^{\mathcal{R}}} Q(u^*(\mathcal{P}), \mathcal{P}; u_{\min}, a) \tag{1}$$

and

$$u^*(\mathcal{P}) \stackrel{\Delta}{=} \arg \max_{u \in D_u(\mathcal{P})} Q(u, \mathcal{P}; u_{\min}, a) \tag{2}$$

where

$$D_u(\mathcal{P}) = \{u : (\forall_{l \in \mathcal{L}} \sum_{r \in \mathcal{R}_l \cap \mathcal{P}} u_r \le U_l) \wedge (\forall_{r \in \mathcal{R} \setminus \mathcal{P}} u_r = 0) \wedge (\forall_{r \in \mathcal{P}} u_{r,\min} \le u_r \le u_{r,\max})\}.$$

So the problem **P2** is now decomposed in two subproblems: discrete subproblem (1) – which refers to an admission control problem (**AC**), and continuous subproblem (2) – which refers to a rate allocation problem (**RA**) for elastic flow, which is widely studied and solutions of such deterministic problems are given e.g. in [6].

## 3. Uncertainty in the computer networks

It must be taken into account that full knowledge on network parameters is usually unavailable due to network dynamics, size, complexity, etc. [7].

Large number of nodes and links can be aggregated into a much smaller number of logical entities. However, as a result, information about the state of individual nodes and

links is often lost. Available bandwidth, associated with the aggregate node are typically obtained by "averaging" corresponding individual metrics. The main consequence of this loss of accurancy in network state information is that it now needs to be consider not only the amount of resources that are available, but also the level of certainty with which these resources are indeed available. [7]

But even there is no aggregation of information, the node and link parameters cannot be assumed to be truly accurate. Typically, they are just approximations of the real parameters and values since they are based on elaborated models that cannot fully represent the complexity of the devices [8].

Another source of uncertainty is existence of hidden information. Interconnected networks may include private networks that hide some or all of their information. Typically, such networks would advertise information that contains inaccuracies or advertise ranges for specific parameters. This information can be interpreted as probability distributions based on parameters supplied by these networks, or by prior experience [8].

Consequence of network dynamics is that many parameters are affected by temporal conditions, such as e.g. congestion. Rapidly advertising the current, updated, accurate parameters can be impractical when the network is highly dynamic and changes are frequent. This implies that parameters values might be based on e.g. average behavior or on worst-case behavior. The precise probability distributions associated with each value depends on a priori knowledge on the frequency of updates and the dynamics of the network [8]. Unlike in wireline networks where links are disjoint resources with fixed capacities, in ad hoc wireless networks the link capacities are "elastic" [9], [10].

Some of these imprecise information can be modeled with assumption that bandwidths ($U_l$) are unknown parameters. The most common approach is based on assumption that those parameters are random variables. However, not always probabilistic approach can be applied, especially when knowledge of imprecise parameters is based on expert's knowledge or a priori assumption as it is proposed in [8]. Then other formalism for modeling uncertainty must be applied – e.g. fuzzy model [11]. In this paper, the formalism of uncertain variables [12], [13] is applied.

## 4. Uncertain variables

In the definition of the uncertain variable $\bar{x}$ we consider two soft properties (i.e. such properties $\varphi(x)$ that for the fixed $x$ the logic value $v[\varphi(x)] \in [0,1]$): "$\bar{x} \cong x$" which means "$\bar{x}$ is approximately equal to $x$" or "$x$ is the approximate value of $\bar{x}$ ," and $\bar{x} \tilde{\in} D_x$" which means "$\bar{x}$ approximately belongs to the set $D_x$ " or "the approximate value of $\bar{x}$ belongs to $D_x$". The uncertain variable $\bar{x}$ is defined by a set of values $X$ (real number vector space), the

function $h(x) = v(\bar{x} \cong x)$ (i.e. the certainty index that $\bar{x} \cong x$, given by an expert) and the following definitions for $D_x, D_1, D_2 \subseteq X$ :

$$v(\bar{x} \,\tilde{\in}\, D_x) = \begin{cases} \max\limits_{x \in D_x} h_x(x) & \text{for} \quad D_x \neq \varnothing, \\ 0 & \text{for} \quad D_x = \varnothing \quad \text{(empty set)}, \end{cases}$$

$$v(\bar{x} \,\tilde{\notin}\, D_x) = 1 - v(\bar{x} \,\tilde{\in}\, D_x),$$

$$v(\bar{x} \,\tilde{\in}\, D_1 \vee \bar{x} \,\tilde{\in}\, D_2) = \max\{v(\bar{x} \,\tilde{\in}\, D_1), v(\bar{x} \,\tilde{\in}\, D_2)\},$$

$$v(\bar{x} \,\tilde{\in}\, D_1 \wedge \bar{x} \,\tilde{\in}\, D_2) = \begin{cases} \max\{v(\bar{x} \,\tilde{\in}\, D_1), v(\bar{x} \,\tilde{\in}\, D_2)\} & \text{for } D_1 \cap D_2 \neq \varnothing, \\ 0 & \text{for } D_1 \cap D_2 = \varnothing. \end{cases}$$

The function $h(x)$ is called a certainty distribution (e.g. triangular certainty distribution is shown in Fig. 2).

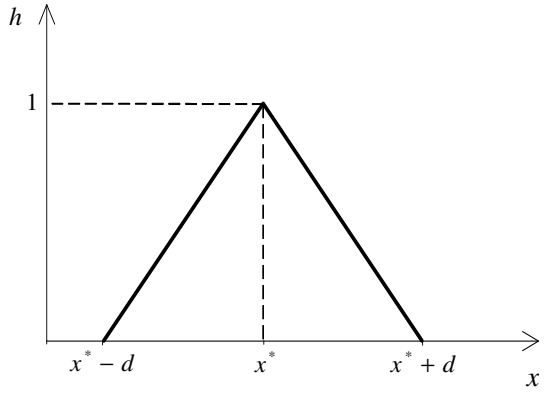



Fig. 2 Triangular certainty distribution

For the uncertain variable one can define a mean value $M(\bar{x})$ in a similar way as expected value for a random variable, in continuous case:

$$M(\bar{x}) = \int_X x\bar{h}(x)dx, \tag{3}$$

where

$$\bar{h}(x) = \frac{h(x)}{\int_X h(x)dx}$$

under the assumptions that respective integrals exist.

Let us consider a static plant with the input vector $u \in U$ and the output vector $y \in Y$, described by a function $y = F(u; x)$ where the vector of unknown parameters $x \in X$ is assumed to be a value of an uncertain variable described by the certainty distribution $h(x)$ given by an expert. For the requirement $y \in D_y \subset Y$ given by a user, we can formulate the following decision problem: For the given $F$, $h(x)$ and $D_y$ one should find the decision $u^*$ maximizing the certainty index that the set of possible outputs approximately belongs to $D_y$ (i.e. belongs to $D_y$ for an approximate value of $x$). Then $u^* = \arg\max_{u \in U} v[y \,\tilde{\in}\, D_y] = \arg\max_{u \in U} \max_{x \in D_x(u)} h(x)$, where $D_x(u) = \{x \in X : y \,\tilde{\in}\, D_y\}$.

## 5. Non-deterministic problem formulations

Because of the uncertainty, bandwidth constraints can be satisfied only in the soft way. It means that one can only determine certainty index of bandwidth constraints satisfying. In such a case, different approaches can be proposed and formulated which lead to different decision making problems (**DMP**). The first one (**DMP$_1$**) consists in the determination of bandwidths using mean values as well as the optimization of the objective function due to QoS and bandwidth constraints imposed. Second possibility is an optimization of the objective function, so that bandwidth constraints are satisfied with certainty index not less than given by the user level $v^*$, is considered (**DMP$_2$**). The third possibility of decision making problem (**DMP$_3$**) is maximization certainty index that bandwidth constraints are approximately satisfied due to QoS constraints and providing the minimal acceptable value of objective function $\alpha$ given by an user. All these approaches have common given data: $\mathcal{R}$, $\mathcal{L}$, $\mathcal{L}_r$ for $r \in \mathcal{R}$, $u_{\min}$, $u_{\max}$, $a$, $Q(u, \mathcal{P}; u_{\min}, a)$, $\mathcal{R}_l$, $h_{U_l}$ for $l \in \mathcal{L}$.

The **DMP$_1$** can be formulated as follows:

**Find:**

$$(\mathcal{P}^*, u^*) = \arg\max_{\mathcal{P} \in 2^{\mathcal{R}}} \max_{u \in M(D_u(\mathcal{P}))} Q(u, \mathcal{P}; u_{\min}, a)$$

where:

$$M(D_u(\mathcal{P})) = \{u : (\forall_{l \in \mathcal{L}} \sum_{r \in \mathcal{R}_l \cap \mathcal{P}} u_r \le M(\overline{U}_l)) \wedge (\forall_{r \in \mathcal{R} \setminus \mathcal{P}} u_r = 0) \wedge (\forall_{r \in \mathcal{P}} u_{r,\min} \le u_r \le u_{r,\max})\}$$

and $M(\overline{U_l})$ is a mean value of uncertain variable $\overline{U_l}$ defined by (3). Once $M(\overline{U_l})$ is determined, **DMP$_1$** is analogical to **P1** and the same algorithms can be applied for solution.

For **DMP$_2$** certainty level $v^*$ must be given in addition, then the decision-making problem can be formulated as follows:

**Find:**

$$(\mathcal{P}^*, u^*) = \arg\max_{\mathcal{P}\in 2^{\mathcal{R}}} \max_{u\in\Delta_u(\mathcal{P})} Q(u, \mathcal{P}; u_{\min}, a)$$

where

$$\Delta_u(\mathcal{P}) = \{u : (\forall_{l\in\mathcal{L}} v[\sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r \tilde{\leq} \overline{U_l}] \geq v^*) \wedge (\forall_{r\in\mathcal{R}\setminus\mathcal{P}} u_r = 0) \wedge (\forall_{r\in\mathcal{P}} u_{r,\min} \leq u_r \leq u_{r,\max})\}. \quad (4)$$

For **DMP$_3$** additionally $\alpha$ value must be given, then the problem can be formulated as follows:

**Find:**

$$(\mathcal{P}^*, u^*) = \arg\max_{\mathcal{P}\in 2^{\mathcal{R}}} \max_{u\in\overline{D}_u(\mathcal{P})} v\left[\wedge_{l\in\mathcal{L}} \sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r \tilde{\leq} \overline{U_l}\right]$$

where

$$\overline{D}_u(\mathcal{P}) = \{u : (Q(u, \mathcal{P}; u_{\min}, a) \geq \alpha) \wedge (\forall_{r\in\mathcal{R}\setminus\mathcal{P}} u_r = 0) \wedge (\forall_{r\in\mathcal{P}} u_{r,\min} \leq u_r \leq u_{r,\max})\}.$$

## 6. Solution algorithms for non-deterministic problems (DMP2 and DMP3)

Assume that for every $U_l$ the certainty distributions are given in the form:

$$h_{U_l} = \begin{cases} \overline{h}_{U_l} & \text{for} \quad U_l^* - d_{U_l} \leq U_l \leq U_l^*, \\ \underline{h}_{U_l} & \text{for} \quad U_l^* \leq U_l \leq U_l^* + d_{U_l}, \\ 0 & \text{otherwise}, \end{cases}$$

where: $\overline{h}_{U_l}$ is the increasing function, $\underline{h}_{U_l}$ is the decreasing function, and $\overline{h}_{U_l}(U_l^* - d_{U_l}) = 0$, $\overline{h}_{U_l}(U_l^*) = \underline{h}_{U_l}(U_l^*) = 1$, $\underline{h}_{U_l}(U_l^* + d_{U_l}) = 0$. The example of such function is triangular certainty distribution shown on Fig. 2.

### 6.1. Solution algorithms for $\mathbf{DMP_2}$

The $\mathbf{DMP_2}$ solution algorithm is as follows:
Let

$$v_l \triangleq v\left[\sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r \,\tilde{\leq}\, \overline{U}_l\right]$$

then

$$v_l = v\left[\overline{U}_l \,\tilde{\in}\, [\sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r, \infty)\right] = \max_{U_l\in[\sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r,\infty)} h_{U_l}.$$

It is easy to notice that certainty index $v_l$ is in the form of

$$v_l = \begin{cases} 1 & \text{for} \quad \sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r \leq U_l^* \quad , \\ \underline{h}(\sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r) & \text{for} \quad U_l^* < \sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r \leq U_l^* + d_{U_l} \, , \\ 0 & \text{otherwise.} \end{cases}$$

Then (4) can be rewritten as:

$$\Delta_u(\mathcal{P}) = \{u : (\underset{l\in\mathcal{L}}{\forall} \underline{h}_{U_l}^{-1}(v^*) \geq \sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r) \wedge (\underset{r\in\mathcal{R}\setminus\mathcal{P}}{\forall} u_r = 0) \wedge (\underset{r\in\mathcal{P}}{\forall} u_{r,\min} \leq u_r \leq u_{r,\max})\}. \qquad (5)$$

For example for triangular certainty distributions, (5) has the form of

$$\Delta_u(\mathcal{P}) = \{u : (\underset{l\in\mathcal{L}}{\forall} \sum_{r\in\mathcal{R}_l\cap\mathcal{P}} u_r \leq U_l^* + (1-v^*)d_{U_l}) \wedge (\underset{r\in\mathcal{R}\setminus\mathcal{P}}{\forall} u_r = 0) \wedge (\underset{r\in\mathcal{P}}{\forall} u_{r,\min} \leq u_r \leq u_{r,\max})\}.$$

One can notice that there is no uncertainty in (5), only parameters characterizing expert's knowledge, so solution methods as in deterministic case (**P2**) can be applied now.

### 6.2. Solution algorithms for $\mathbf{DMP_3}$

The $\mathbf{DMP_3}$ can be also decomposed similarly to deterministic case (**P2**), which means that for every possible set $\mathcal{P}$ one should find the rate allocation which maximizes the

certainty index: $v^*(\mathcal{P};\alpha) \triangleq \max_{u \in \overline{D}_u(\mathcal{P})} \min_l v_l$ (**RA** subproblem) and than one should find such set $\mathcal{P}$ for which the solution of **RA** subproblem gives the greatest certainty index value.

The solution algorithm of **RA** subproblem of **DMP$_3$** is the following:

If there exists any rate allocation for which certainty index is equal 1, it is optimal solution of **RA** subproblem of **DMP$_3$**, because the maximal value of certainty index cannot be greater than 1. To check this, one should determine if set $\hat{D}_u(\mathcal{P}) = \{u : u \in \overline{D}_u(\mathcal{P}) \wedge \underset{l \in \mathcal{L}}{\forall} v_l = 1\}$ is empty or not and it can be easily done by finding $u^*_{\mathrm{DMP}_1}(\mathcal{P})$ i.e., the optimal solution of **RA** subproblem of **DMP$_1$** (assuming $M(\overline{U}_l) = U_l^*$) if it exists. If $Q(u^*_{\mathrm{DMP}_1}(\mathcal{P}), \mathcal{P}; u_{\min}, a) \geq \alpha$ then $u^*_{\mathrm{DMP}_3}(\mathcal{P}) = u^*_{\mathrm{DMP}_1}(\mathcal{P})$ is also an optimal solution of **DMP$_3$** and $v^*(\mathcal{P};\alpha) = 1$. If $\hat{D}_u(\mathcal{P})$ is empty the maximal value of certainty index is less than 1. Then let $u^*_{\mathrm{DMP}_2}(\overline{v}, \mathcal{P})$ be the optimal solution of **RA** subproblem of **DMP$_2$** depending of $\overline{v} \in D_v$ and $Q(u^*_{\mathrm{DMP}_2}(\overline{v}, \mathcal{P}), \mathcal{P}; u_{\min}, a) \triangleq Q^*_{\mathrm{DMP}_2}(\overline{v}, \mathcal{P})$, where $D_v = \{v \in [0,1) : \underset{l \in \mathcal{L}}{\forall} \sum_{r \in R_l} u_{r,\min} \leq \underline{h}^{-1}_{U_l}(v)\}$ is a set of certainty thresholds for which the feasible solution exists. It can be shown that $Q^*_{\mathrm{DMP}_2}(\overline{v}, \mathcal{P})$ is non-increasing function of $\overline{v}$, so the solution of **RA** subproblem of **DMP$_3$** can be obtained as $u^*_{\mathrm{DMP}_3}(\mathcal{P}) = u^*_{\mathrm{DMP}_2}(\overline{v}(\mathcal{P};\alpha), \mathcal{P})$, where $\overline{v}(\mathcal{P};\alpha)$ is the solution of $Q^*_{\mathrm{DMP}_2}(\overline{v}, \mathcal{P}) = \alpha$, if it exists and then $v^*(\mathcal{P};\alpha) = \overline{v}(\mathcal{P};\alpha)$. If there does not exist any $\mathcal{P}$ for which $\overline{v} \in D_v$ can be determined from $Q^*_{\mathrm{DMP}_2}(\overline{v}, \mathcal{P}) = \alpha$, there are two possibilities: if $Q^*_{\mathrm{DMP}_2}(\sup D_v, \mathcal{P}) \geq \alpha$ then $v^*(\mathcal{P};\alpha) = \sup D_v$ and $u^*_{\mathrm{DMP}_3}(\mathcal{P}) = u^*_{\mathrm{DMP}_2}(\sup D_v, \mathcal{P})$ else for every rate allocation $u$ certainty index $v\left[\underset{l \in \mathcal{L}}{\wedge} \sum_{r \in \mathcal{R}_l \cap \mathcal{P}} u_r \tilde{\leq} \overline{U}_l\right] = v^*(\mathcal{P};\alpha) = 0$ and every rate allocation in $\overline{D}_u(\mathcal{P})$ is feasible.

The optimal solution $\mathcal{P}^*$ of **AC** subproblem is the one for which $v^*(\mathcal{P};\alpha)$ is the greatest.

## 7. Numerical example

Let us consider a numerical example with the network utility function in the form of $Q(u,\mathcal{P};u_{\min},a)=\sum_{r\in\mathcal{P}}a_r^{(1)}u_r-\sum_{r\in\mathcal{R}/\mathcal{P}}a_r^{(2)}u_{r,\min}$. Assume the network topology as in Fig. 3 and the following numerical data: $R=3$, $u_{r,\min}=1$ for $r=1,2,3$, $a_1^{(1)}=a_2^{(1)}=2$, $a_3^{(1)}=3$, $a_r^{(2)}=1.5$ for $r=1,2,3$ and triangular certainty distributions parameters $U_1^*=1.8$, $d_{U_1}=1$, $U_2^*=1.8$, $d_{U_2}=1$, $U_3^*=2$, $d_{U_3}=1$. Let $\bar{v}=0.9$ and $\alpha=7$ be user requirement in **DMP$_2$** and **DMP$_3$** respectively.

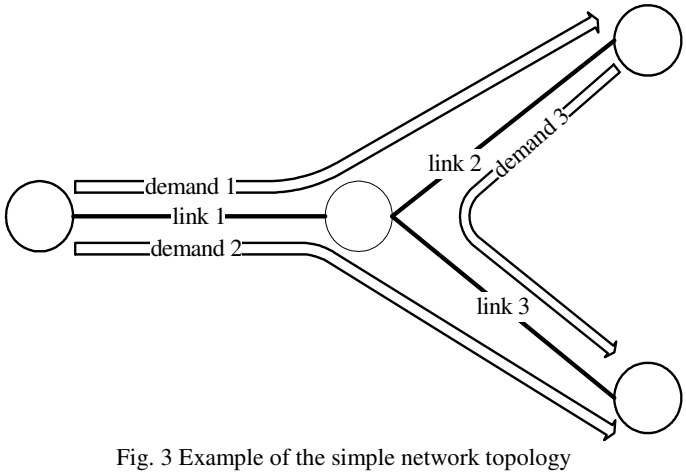



Fig. 3 Example of the simple network topology

The results for three introduced non-deterministic decision-making problems are as follows:

**DMP$_1$:** $\mathcal{P}^*=\{2,3\}$, $u_1^*=0$, $u_2^*=1$, $u_3^*=1$, $Q^*_{\mathrm{DMP}_1}(\mathcal{P})=2.5$.

**DMP$_2$:** $\mathcal{P}^*=\{2,3\}$, $u_1^*=0$, $u_2^*=1.05$, $u_3^*=1.05$, $Q^*_{\mathrm{DMP}_2}(\mathcal{P})=2.7$.

**DMP$_3$:** $\mathcal{P}^*=\{1,2,3\}$, $u_1^*=1$, $u_2^*=1$, $u_3^*=1$, $v^*=0.8$, $Q^*_{\mathrm{DMP}_3}(\mathcal{P})=7$.

All three non-deterministic decision-making problem formulations proposed in the paper can lead to different rate allocations. Differences in the problem formulations implies different possible application to the practical issues. **DMP$_1$** is the simplest to solve, but the least of expert's knowledge is used. In the **DMP$_2$** we use expert's knowledge of the unknown parameter more efficiently, but there must be certainty threshold specified additionally. In the **DMP$_3$** it is assumed that utility threshold $\alpha$ must be given. It can be justified especially in offline algorithm when economic interpretation of utility can be given or $\alpha$ may be determined by an expert.

## 8. Final remarks

In this paper applying uncertain variables to the rate allocation problem in the computer network with unknown parameters was proposed. Three non-deterministic decision-making problems were formulated and the solution algorithms were given. The numerical example was presented and commented.

The effective determining of $\mathcal{P}$, a set of admitted transmission demands, as well as C-uncertain variables application for description of uncertainty are still worth considering.

## 9. Acknowledgement

The research was supported in 2005-2007 by the Ministry of Science and Higher Education under Grant No. 3 T11A 031 28.

# References

[1] Kelly F.P., Maulloo A.K., Tan D.K.H., Rate control for communication networks: shadow prices, proportional fairness, stability, Journal of the Operational Research Society, vol. 49, 1998

[2] La R., Anantharam A.,Utility-Based Rate Control in the Internet for Elastic Traffic, IEEE Transactions on Networking, Vol. 10, No. 2, 2002

[3] Vikram S., Chiasserini C.-F., Nuggehali P.S., Rao R.R., Optimal Rate Allocation for Energy Efficient Multipath Routing in Wireless Ad Hoc Networks, IEEE Transactions on Wireless Communications, Vol. 3, No. 3, 2004

[4] Mitra D., Wang Q., Stochastic Traffic Engineering for Demand Uncertainty and Risk-Aware Network Revenue Management, IEEE Transactions on Networking, Vol. 13, No. 2, 2005

[5] Chatterjee M., Lin H., Das S.K., Rate Allocation and Admission Control for Differentiated Services in CDMA Data Networks, IEEE Transactions on Mobile Computing, Vol. 6, No. 2, 2007

[6] Palomar D.P., Chiang M., A Tutorial on Decomposition Methods for Network Utility Maximization, IEEE Journal on Selected Areas in Communications, Vol. 24, No. 8, 2006

[7] Guerin R.A., Orda A., QoS Routing in Networks with Inaccurate Information: Theory and Algorithms, IEEE Transactions on Networking, Vol. 7., No. 3, 1999

[8] Lorenz D.H., Orda A., QoS Routing in Networks with Uncertain Parameters, IEEE/ACM Transactions on Networking, Vol. 6, No. 6, 1998

[9] Chen L., Low S., Chiang M., Doyle J., Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks

[10] Zhang G., Wu Y., Liu Y, Stability and Sensitivity for Congestion Control in Wireless Networks with Time Varying Link Capacities

[11] Drummond A.C., da Fonseca N., Devetsikiotis M., Yamakami A., Bandwidth Allocation in Self-Sizing Networks Under Uncertain Constraints, Proceedings of IEEE International Conference on Communications, Istanbul 2006

[12] Bubnicki Z., Analysis and Decision Making in Uncertain Systems, Springer, Berlin, London, New York 2004

[13] Bubnicki Z., Uncertain Logics, Variables and Systems, Springer, Berlin, London, New York 2002

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 281–292

# VoIP traffic modelling in a multimedia gateway

ARKADIUSZ BIERNACKI [a]

[a]Institute of Computer Science
Silesian University of Technology
arkadiusz.biernacki@polsl.pl

**Abstract:** In this work we propose three multiplexed VoIP traffic models. The models take into account burst and connection (session) scales. The models are based on Markov processes. The parameters of the models are estimated from the measurements of VoIP conversation characteristic or are computed directly from multiplexed VoIP traffic. We used the models to analyse the queuing behaviour of a multimedia gateway buffer.
**Keywords:** Computer network performance, Integrated voice-data communication, Markov processes, Modelling

## 1. Introduction

The growth of communication based on Voice over IP protocol (VoIP) has been exceptional during recent years and is expected to continue in the future. Consequently, voice packets produced during telephone conversations are to have considerable share in all voice packets sent through networks. When certain amount of voice calls is performed simultaneously on a single link, the link needs to be shared between them, and a statistical multiplexing of voice packets is necessary. The multiplexing process is usually performed by a multimedia gateway which resides in a border between the traditional telecommunication network and computer network transporting VoIP packets. The gateway performs time division multiplexing (TDM), where periodically one user at time gains control of a full capacity of a link for a short instance of time.

The ratio of the number of VBR sources that can be multiplexed on a fixed capacity link under a specified delay or loss constraint to the number of sources that can be supported on the basis of peak rate allocation is called statistical multiplexing gain (SMG). To determine and maximise the SMG, admission control rules are formulated that relate to traffic characteristics, which flow into the buffer of multimedia gateway, the gateway performance constraints and parameters. To formulate these rules a multiplexed traffic model is needed. This topic is a main concern of our work.

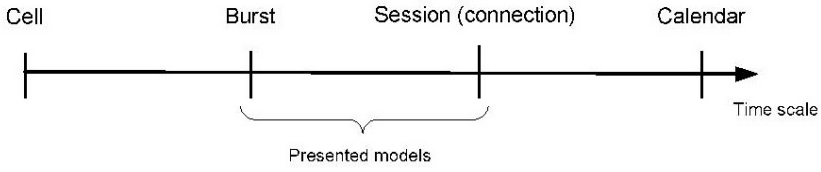



Fig. 1: Levels of traffic behaviour

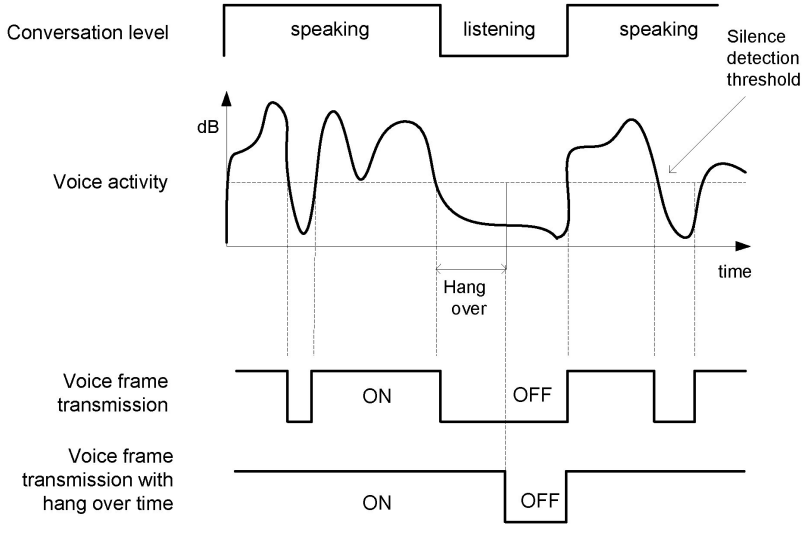



Fig. 2: The correspondence between conversation levels and a packet generation by a voice coder – the burst scale

Two different types of VoIP traffic behaviour are being modelled. In the first case the behaviour concerns the use made of the telephony service by customers (in terms of how often the service is used, and for how long) – the session (connection) scale. In the second case, the focus is at the characteristic of telephony behaviour – the burst scale, Fig. **??**. A VoIP conversation, like traditional telephony conversation, can be considered an alternate process during which one of interlocutors is speaking and the other one is listening. Moreover, the speaking interlocutor often makes small gaps between spoken phrases, words and syllables. The aforementioned characteristics of a conversation are used by a voice coder with voice activity detector (VAD), which detects frames containing silence and suspends them from further emission through packet network, Fig. **??**

Multimedia gateway can be considered a kind of statistical multiplexer [**?**] thus it is usually modelled as queuing systems with buffer space of size $B$, to which maximum of $N$ variable bitrate (VBR) sources are connected, served by a transmission link of fixed capacity

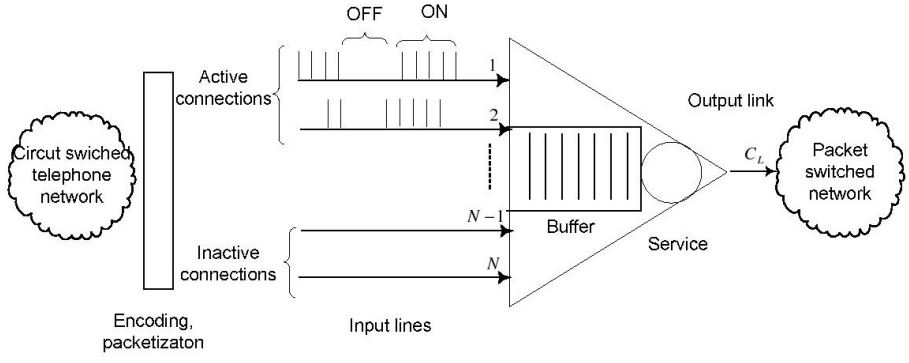



Fig. 3: Multimedia gateway model

$C_L$, Fig. **??**

## 2. Theoretical background

Markov Modulated Process (MMP) is described by matrix $\mathbf{Q} = [q_{ij}]$ (infinitesimal generator of Markov chain $X(t)$) where $q_{ij}$ are transition coefficients, and matrix

$$\mathbf{R} = \begin{pmatrix} r_1 & 0 & \dots & 0 \\ 0 & r_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r_S \end{pmatrix}, \tag{1}$$

where $r_i$ coefficients describe traffic intensity generated when Markov chain is in state $i$, i.e. $X(t) = i$, Fig. **??** [**?**], chapter 4.

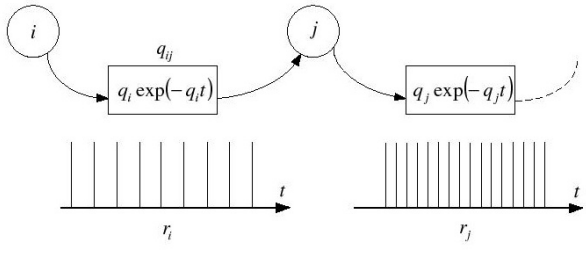
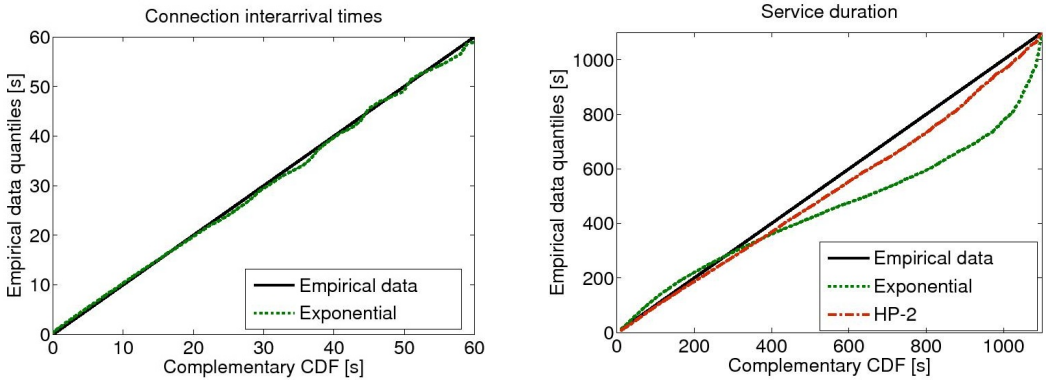



Fig. 4: Continuous time MMP




(a) Connection interarrival times

(b) Service duration

Fig. 5: Quantile-quantile plot for the distribution of the connection interarrivals times and service times

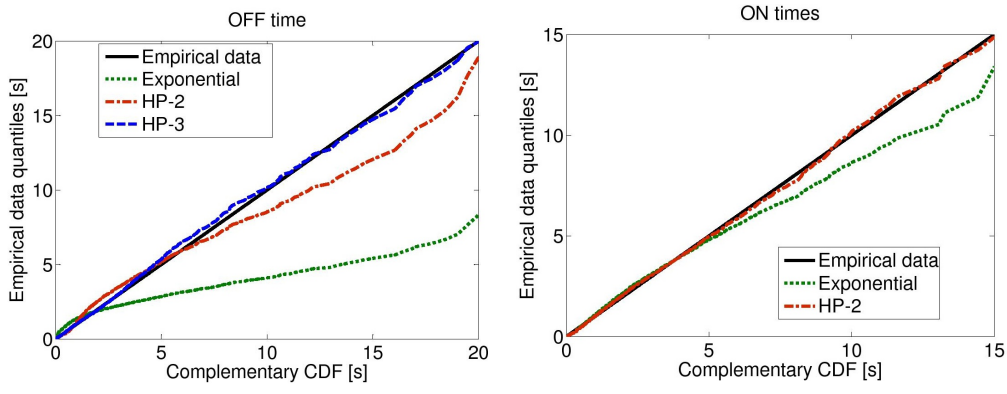
## 3. Data acquisition and analysis

### 3.1. Connection level modelling

The data used in the analysis were derived from the main telephone exchange of the Silesian University of Technology. It contained the record of about eighty thousand connections recorded in December 2005 using traditional telephone lines. The record included the beginning time of a connection and its duration time with one-second accuracy. Having excluded the data from holidays and weekends, we analyzed the set generating between 10-14 o'clock, which was a homogenous arrival Poisson process not influenced by time dependencies, Fig. **??**. We also stated that connection duration times slightly diverge from an exponential distribution, Fig. **??**.

### 3.2. Burst level modelling

We studied the ON/OFF times distribution of voice streams produced by a G.711 coder equipped with a VAD. With the use of Windows Sound Recorder we recorded one side of several real phone conversations held using popular VoIP software. We connected two computers equipped with OpenH323 library [**?**] with Ethernet cable. Previously recorded conversations were played and encoded by voice coders. On the basis of the recorded timestamps we calculated the ON/OFF time periods. The detailed analysis of G.711 coder showed that, taking into account analytically traceable distributions, the ON/OFF periods are optimally approximated by two- and three-stage hyperexponential distributions respectively (HP-2 and HP-3), Fig. **??**.




(a) OFF times (b) ON times

Fig. 6: Quantile-quantile plot for the ON/OFF times adjusted to hyperexponential distributions

## 4. Proposed models

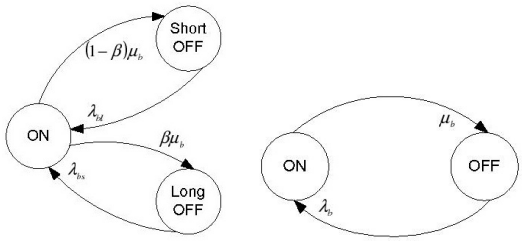
We propose three models, Model 1, Model 2 and Model 3, based on the Markov Chain. The models act at two time scales – the connection and the burst one. The connection scale describes conversations durations and their inter-arrival times. Connections are generated according to the Poisson process with rate $\lambda_c$. Their durations are exponentially distributed with a mean $\mu_c^{-1}$. Values for the above coefficients were obtained as results of analysis presented in section **??**

Number of VoIP source states, which the models are based on, is a compromise between models accuracy and their complexity. Model 1 is based on the two-state VoIP source, Fig. **??**. For each connection, number of flows forms a Markov birth-death process [**?**] and the maximum flows number depends on the current number of connections. The transition rates between flows are based on $\mu_b$ and $\lambda_b$ constants and additionally depend on the number of connections. The structure of Model 1 was presented on Fig. **??**.

The base of Model 2 is the three-state VoIP source, presented on Fig. **??**. Every state of the model consists of the connections number, the number of sources being in OFF states and the number of sources being in short OFF states. The transition rates between flows are based on $\mu_b$, $\lambda_{bl}$ and $\lambda_{bs}$ constants. Additionally, they also depend on the number of connections. By $\beta$ we denote the relative frequency of the long OFF states occurrence in the all OFF states. The structure of Model 2 was presented on Fig. **??**.

Values for the above burst scale coefficients were obtained as results of analysis presented in section **??**

The total number of states in the above models is given by $\sum_{i=0}^{N} \binom{i+j-1}{i}$, where $N$ is the number of multiplexed lines (VoIP sources) and $j$ is the number of states in the elementary VoIP source. Hence, the first model is suitable for approximation of a few dozens of




(a) Three-state (b) Two-state

Fig. 7: VoIP source model based on the Markov Chain

simultaneous connections. Because of combinatorial explosion of the number of states in the second model, it is rather suitable for modelling of multiplexed traffic originating from a smaller number of connections. Its advantage is better accuracy in comparison to Model 1.

Parameters of Model 3 are estimated from synthetically generated traffic trace (described in section **??**). The parameter estimation procedure we used was described in detail in [**?**], where this method have been successfully applied for Internet traffic modelling. The model does not explicitly take into consideration elementary VoIP sources presented on Fig. **??**. Model 3 is less susceptible for the combinatorial explosion of number of states hence it can be used to approximate traffic from greater number of sources compared to Model 1 or Model 2.

## 5. Models validation

We validated our models against synthetically generated traces, produced by G.711 voice coder, which flow into multimedia gateway equipped in $N$ input telephone lines (the maximum number of multiplexed VoIP sources). The data used in the validation was described in section **??**

In order to get heavier or lighter multiplexed VoIP traffic we manipulated the arrival rate. However, Poisson property of the arrival process was maintained. We sent recorded conversations through the network to the second computer where we measured the timestamps of the voice packets using Ethereal software [**?**]. We obtained single binary time series, where 0-values corresponded to OFF periods and 1-values corresponded to ON periods. From these values, for each starting connection, a set was randomly chosen. Its length equalled the connection duration time. For the all active connections we were totalling up the values of the sets in discrete periods obtaining the time series which represented the traffic intensity, Fig. **??**.

On the example figures **??**–**??** we presented evaluation of a mean, variation, autocorrelation, Hurst $H$ parameter of traffic intensity approximated by the models and the simulation. It is hard to compare each model with others, because the models are dedicated for different

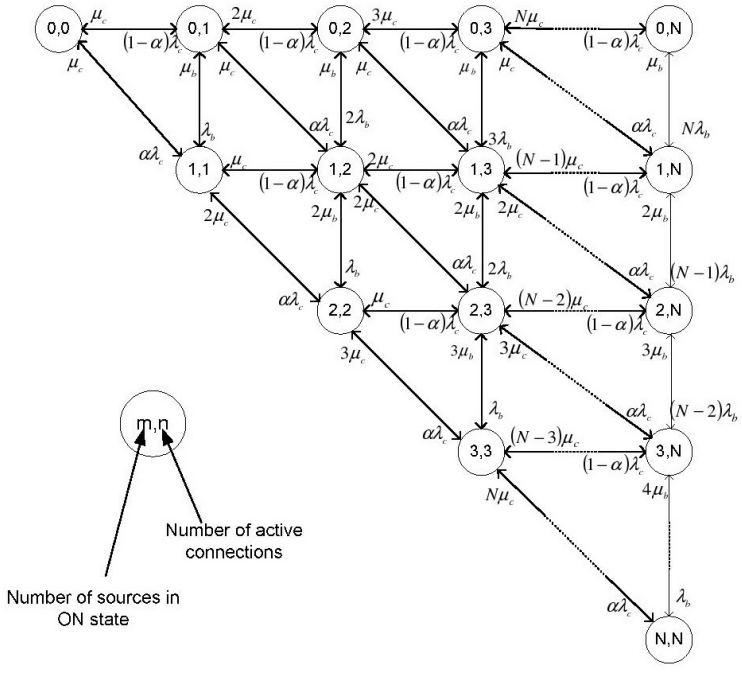



Fig. 8: Multiplexed VoIP traffic model based on the two-state VoIP source

values of $N$.

## 6. Multimedia gateway performance modelling

We used our models to analyse queuing behaviour of multimedia gateway buffer using the fluids models theory. In these models, fluid flows into a fluid reservoir according to a stochastic process. In our case, fluid buffer was either filled or depleted, or both, at rates which are determined by a state of a background Markov process.

Let $Y(t)$ denote the amount of fluid at time $t$ in a reservoir. Furthermore, let $(X(t), R_X)$ be a MMP. $X(t)$ is said to evolve "in the background". The content of the reservoir $Y(t)$ is regulated in such a way that the net input rate into the reservoir (i.e. the rate of change of its content) is $\Delta r_i = r_i - C_L$ at times when $X(t)$ is in state $i$. Hence we have:

$$\frac{dY(t)}{dt} = \begin{cases} 0 & \text{if } Y(t) = 0 \text{ and } \Delta r_X < 0, \\ \Delta r_{X(t)}. \end{cases} \tag{2}$$

The stability condition is given, $\sum_{i \in S} \pi_i \Delta r_i < 0$ , where $\pi_i$ is a stationary probability that $X(t)$ is in state $i$. When the stability condition is satisfied $Y(t)$ converges in distribution as $t \to \infty$. Hence, the stationary joint distribution of $X(t)$ and $Y(t)$ exists and is given by

$$F_i(y) = P[X = i, Y \leq y], \quad i \in S,\ y \geq 0.$$

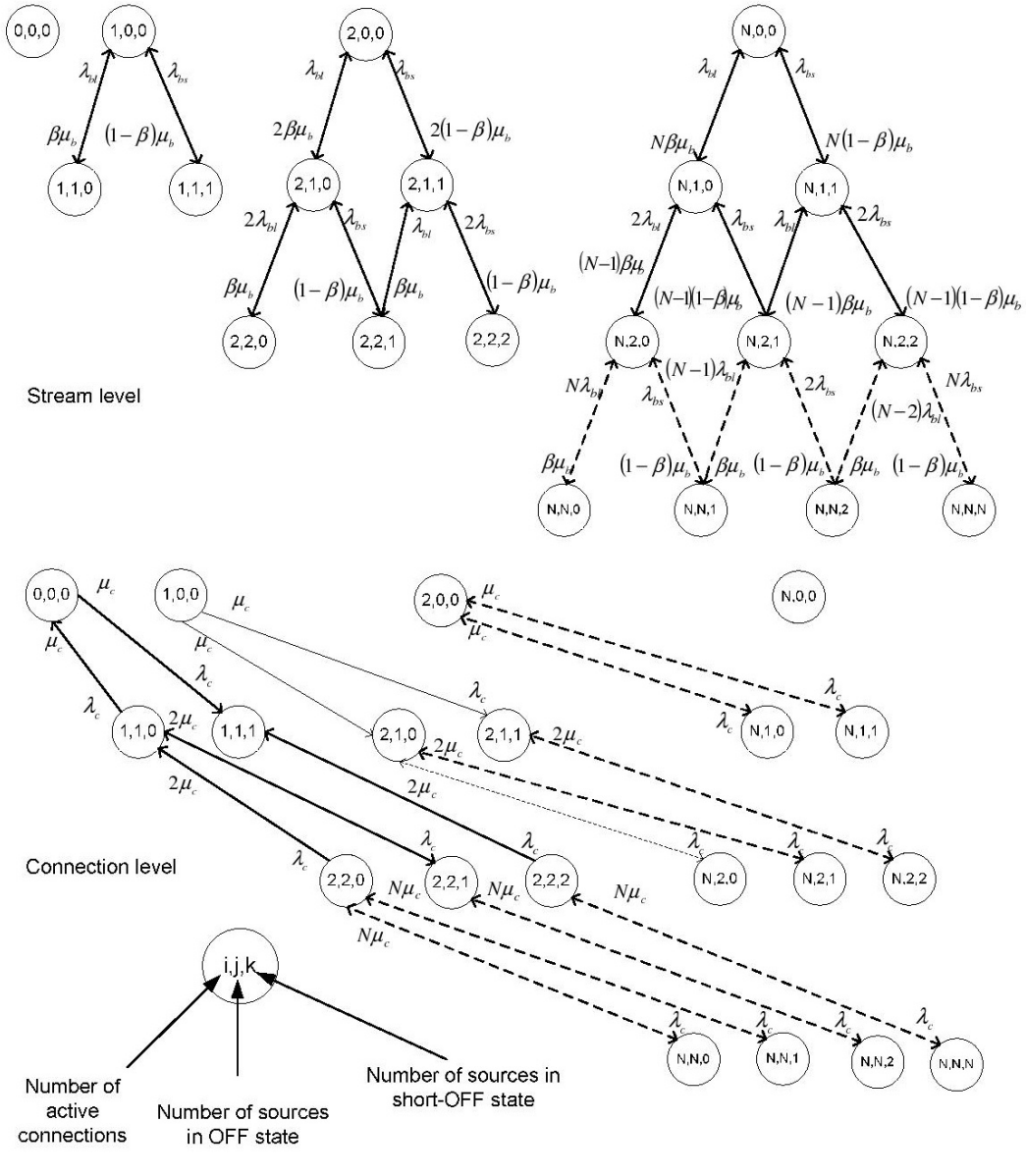



Fig. 9: Multiplexed VoIP traffic model based on the three-state VoIP source

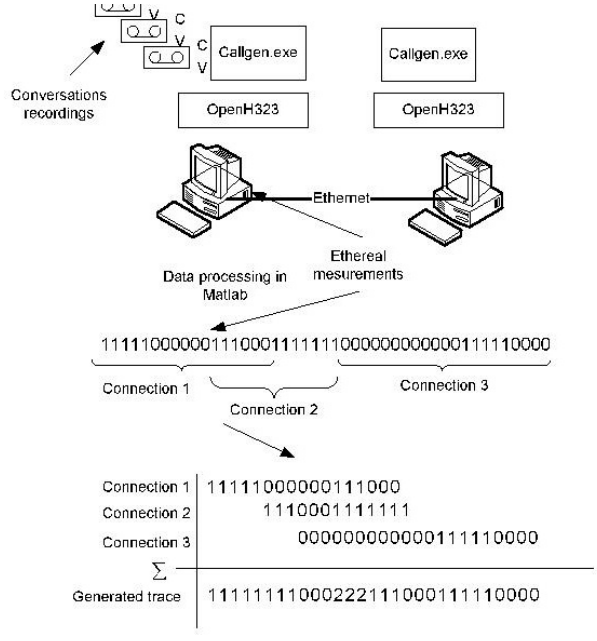
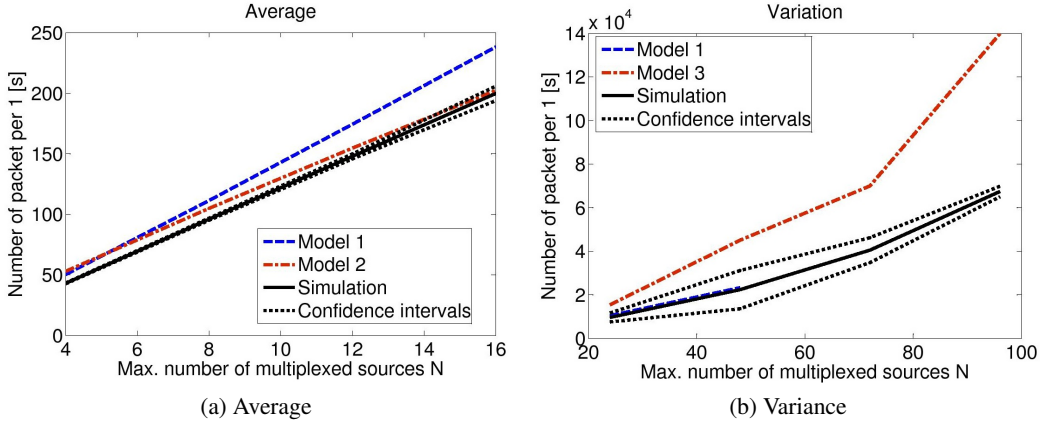
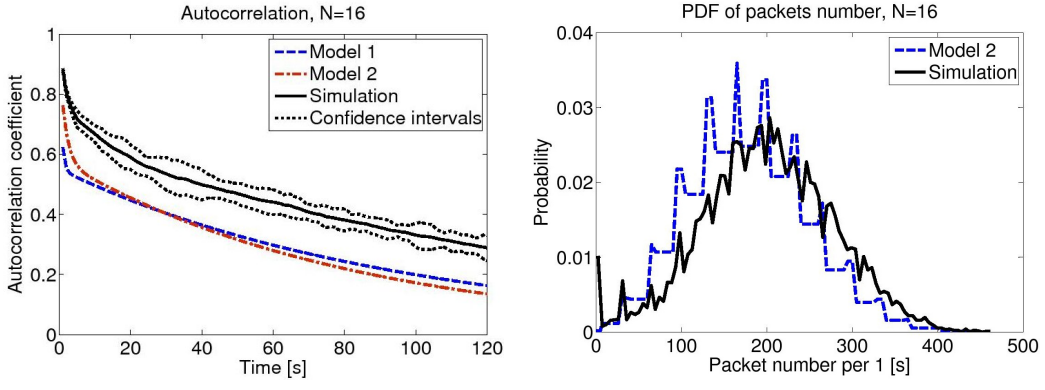



Fig. 10: The generation process of simulated VoIP traces



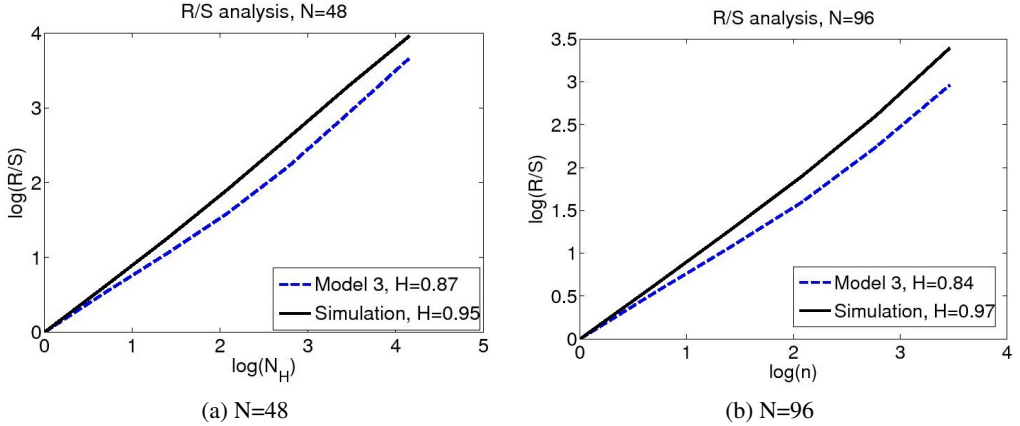

(a) Average (b) Variance

Fig. 11: Comparison of average and variance of traffic intensity approximated by the models and the simulation




(a) Autocorrelation (b) Probability distribution function

Fig. 12: Comparison of autocorrelation and PDF of traffic intensity approximated by the models and the simulation




(a) N=48 (b) N=96

Fig. 13: Comparison of Hurst parameter of traffic intensity approximated by the models and the simulation

It can be shown that the vector $\mathbf{F}(y) = [F_1(y), \ldots, F_n(y)]^T$ satisfies the differential equation

$$\mathbf{F}'(y)\Delta\mathbf{R} = \mathbf{F}(y)\mathbf{Q}, \tag{3}$$

where prime denotes differentiation. $\Delta\mathbf{R}$ is a diagonal matrix $\Delta\mathbf{R} = \mathrm{diag}(\Delta r_1, \ldots, \Delta r_S)$, $\mathbf{Q}$ is the generator of the Markov process $X(t)$ of size $S \times S$. By assuming that $\Delta\mathbf{R}$ is non-singular i.e. $\forall i \in S\ \Delta r_i \neq 0$ and the eigenvalues are simple, it follows that the solution of (**??**) is given by

$$\mathbf{F}(y) = \sum_{i=1}^{S} c_i e^{\xi_i y}\mathbf{v}_i, \tag{4}$$

where the $(\xi, \mathbf{v})$ are the eigenvalue-eigenvector pairs of the matrix $\mathbf{Q}\Delta\mathbf{R}^{-1}$ and $c_i$ are constants that can be determined by boundary conditions. Further details of the above method can be found in [**?**], pages 13–14 and 30–31.

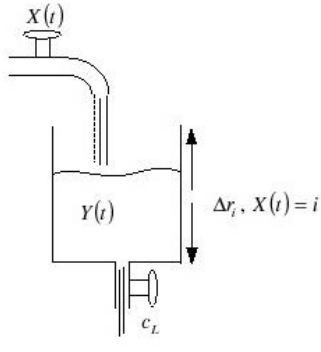



Fig. 14: Graphical interpretation of Markov fluid model

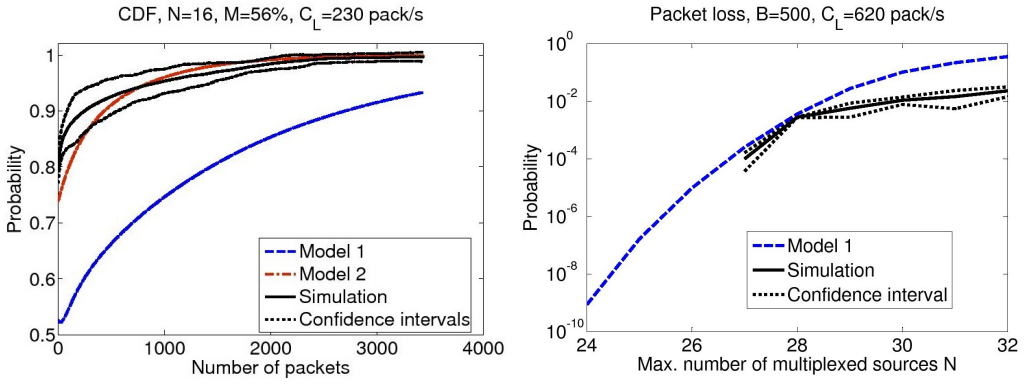
Probability of the buffer overflow can be estimated as [**?**]

$$P_B \approx P\{Y > B\} = 1 - F(B), \tag{5}$$

where $B$ is size of the buffer. Example results of the gateway buffer analysis are presented on Fig. **??**. Variable $M$ denotes relation between the gateway output link bandwidth $C_L$ and a maximum traffic intensity $P_{\max}$ generated by $N$ sources, i.e. $C_L = M\%P_{\max}$.

## 7. Conclusion

In this paper we studied the multiplexed VoIP traffic models and the multimedia gateway performance. We proposed two models, based on the continuous-time Markov Chain. The models differ in their complexity and accuracy. Model 1 and Model 2 are more flexible but also susceptible for the combinatorial explosion of the number of states. Model 3 is less




(a) CDF of packets number

(b) Buffer overflow

Fig. 15: Cumulative distribution of packets number in a multimedia gateway buffer and probability of the buffer overflow

accurate but allows for approximation of a greater number of VoIP sources. We showed that the proposed models can approximate with fair accuracy multiplexed voice traffic by computing and evaluating its first and second order statistics. We used the models to analyse the queuing behaviour of multimedia gateway buffer with fair results.

## References

[1] T. Czachórski. *Modele kolejkowe w ocenie efektywności pracy sieci i systemów komputerowych (in Polish only)*. Wydawnictwo Politechniki Śląskiej, Gliwice, Poland, 1999.

[2] Y. Hyun-Kyung and K. Byung-Ryong. A media stream processing of VoIP media gateway. In *The 9th Asia-Pacific Conference on Communications*, volume 1, pages 91–94, 2003.

[3] Ethereal Software Inc. Ethereal, a network protocol analyzer, 2005.

[4] A. Nogueira, P. Salvador, R. Valadas, and A. Pacheco. Fitting self-similar traffic by a superposition of MMPPs modeling the distribution at multiple time scales. *IEICE Transactions on Communications*, Vol. E87-B, No. 3:678–688, 2004.

[5] Z. Papir. *Ruch telekomunikacyjny i przeciążenia sieci*. WKiŁ, 2001.

[6] W. Scheinhardt. *Markov-modulated and feedback fluid queues*. Ph.d. thesis, University of Twente, the Netherlands, 1998.

[7] J. Virtamo. Overflow in a buffered system: fluid queues - materiały do wykładów.

[8] Vox-Gratia. OpenH323, 2005.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 293–300

# Parallel simulation of networks with packet loss*

MATEUSZ NOWAK SŁAWOMIR NOWAK

Institute of Theoretical and Applied Informatics
Polish Academy of Science
ul. Bałtycka 5, Gliwice, Poland
{m.nowak|emanuel}@iitis.gliwice.pl

**Abstract:** Parallel simulation performance strongly depends from synchronisation method and model parameters. Therefore synchronisation algorithms for specific model types should be developed. In the paper parallel simulations, executed in OMNeT++ simulator along with INET extension for TCP/IP networks, were showed, along with performance results. The concept of parallel event-driven simulator of computer networks with non-zero rate of packet loss (e.g. wireless networks) was also shown in the paper.

**Keywords:** network simulations, parallel simulations, wireless networks, OMNeT++

## 1. Introduction

Computer networks simulation is an important tool, essential for investigation of network properties. It is used in order to confirm analytical traffic models, in research on new technologies and protocols, and also in forecasting of phenomena occurring in networks, especially in large and non-typical topologies. Computer networks simulators are based on discrete events simulation technique. They can be evaluated from the point of view of simulation quality, amount and accuracy of library models, and also simulation performance. Simulation experiments, in spite of continuous increase of processors performance, are still time-consuming. One of directions of simulation technique development is looking for new more effective simulation methods, allowing shortening of simulation duration.

The OMNeT++ event-driven simulator ([11]) is used in IITiS PAN as the basic tool in networks simulation research. Its main advantages are simple process of simulation models creation, convenient interface and rich libraries. Specifically INET package ([5]), which is also widely used for Internet simulations, defines a lot of protocols used in Internet networks. Beyond TCP and IP there is UDP, Ethernet, PPP and MPLS with LDP and RSVP-TE signalling.

* Supported by MNiSW grant no. N517 025 31/2997

## 2. Parallel simulation

The event-driven simulation is easy parallelisable. In such a simulation every event is described by a simple data structure. There are no fundamental hardness with construction of parallel discrete events simulator (PDES), sending these structures over the network([3, 8]). Notwithstanding, there are some issues to solve.

Main problem is synchronisation. In parallel simulation every local process (LP) stores its own simulation clock, and its own event list, which is processed independently of other LP's. It can happen, that LP receives message about event occurring in time lesser than current simulation clock value (straggler message). Attempt to process such a message would be break of causality constraints and would lead to erroneous simulation. To prevent such a situation, synchronisation algorithms were worked out.

Conservative synchronisation algorithms, like Null Messages ([1, 2]) prevent LP to progress with simulation clock unless it is sure it would not receive straggler message. Conservative methods are simple to implement, but they have low performance, as lot of time is consumed for exchanging synchronisation messages (called Null Messages) among LP's.

Time Warp synchronisation algorithm ([6, 7]) is basic algorithm in group of optimistic synchronisation methods. LP does not wait for other nodes to synchronise. In the case of receiving straggler message, simulation state and clock are rolled back to the state from before message time stamp. Rolling back a simulation on single LP usually leads to roll-backs also on other nodes. Optimistic algorithms work in general with better performance than conservative ones, but they require a lot of memory for remembering past simulation states, and simulation roll-backs are time-costly.

OMNeT++ simulator includes support for parallel simulation. It allows LP's to communicate in several ways (basic methods are Named Pipes for shared memory multiprocessors and MPI for distributed systems). Synchronisation is accomplished with Null Messages.

INET library is very useful in simulation of TCP/IP networks, but it was not prepared with parallel simulation in mind. Presented work embraced preparation of INET version adapted in limited way to use in parallel simulation.

## 3. Further work motivation

Simulation tasks often require a lot of computing power, what involves their long duration. In necessity of carrying out lots of time-consuming simulation research, time of their execution becomes particularly significant.

Simulation acceleration can be achieved by simultaneous use of many processors. Also in the case of simulation of big network topologies, embracing thousands of nodes, parallel use of many computers could be profitable, as it allows to hold entire model in primary storage, avoiding use of swap files. From the other side, multi-processor computers (including multi-cores processors) and computing clusters are still more available. This motivates further work on parallel simulation methods. As shown below, there is still big demand for

efficient simulators, especially addressing specific problems of wireless mobile networks.

## 4. Synchronisation

As shown in section 4., main difficulty with improving simulation performance is necessity of synchronisation. Known methods are either slow due to communication demands or not effective due to memory requirements. A few experiments, showing influence of synchronisation mechanisms on parallel simulation performance were conducted.

For simulation experiments we used network topology as shown on Fig. 1. Properties of this network, were examined in research on traffic self-similarity ([10]). Network consists of two segments, each containing ten hosts and router. Routers are connected with PPP link of capacity 1,5 Mb/s. Every host can be server or client. Hosts and servers communicate with TCP protocol. Clients require answers of given size and server responds on these requisitions.

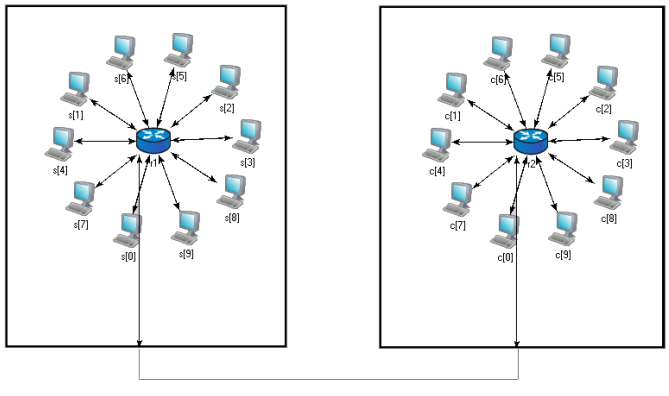



Fig. 1. Simulated network topology

Two variants of topology, with the same physical scheme were used in experiments:

A: All host in segment 1 are clients, directing their requests to servers in segment B. All traffic passes through PPP link, which is bottle-neck of the network, as it shares 10 TCP connections.

B: In every segment 5 hosts are clients and remaining 5 are hosts. 3 TCP connections are directed from clients to servers in the same segment, remaining 2 – to another segment. PPP link between router is shared by 4 TCP connections.

In parallel simulation each segment was simulated by another logical process. Experiments were conducted with OMNeT++/INET on following hardware configurations:

- PC1 – single workstation with single-core AMD Sempron 3400 processor, 512 MB RAM, Windows XP
- PC2 – single workstation with dual-core processor Intel Core2, 3GHz, Windows Vista
- PC3 – two virtual machines, each running Linux Fedora and OpenMPI, hosted on Windows workstations.

For results comparison simulations were conducted in parallel and serial version. In serial simulation entire network was modelled by single simulation process, in parallel one segment A and segment B were placed in different LP's.

In shared memory configuration running under control of MS Windows (PC2), Named Pipes were used as communication layer. In PC3 LP's communicated with each other with OpenMPI. The experiments parameter was link delay on PPP connection. In Null Messages algorithm used in OMNeT the bigger is delay, the more seldom are synchronisation messages and lesser influence of synchronisation on simulation performance. Performance (measured in simulation seconds per minute of wall-clock time) appeared very sensitive for changes of this value.

The dependence between simulated time elapsed during 60 s processor time and link delay is shown on Fig. 2. Simulation speed is the reciprocal of simulation time elapsed during given physical (wall-clock) time. Simulations were performed in serial and parallel configuration. Fig. 3 illustrates acceleration, defined as proportion of simulation time achieved for serial and parallel simulation, or between parallel and serial simulation speed. Result close to 2 means almost linear acceleration achieved on two processors (cores).

Results can be summarised as follows:

1. For PC2 configuration with dual-core processor for big enough Link Delay value $(> 0,4)$ parallel simulation acceleration is better than 1 (parallel simulation is faster than serial). For low Link Delay values influence of Null Messages is too high to achieve any speed-up, but for bigger ones trends asymptotically to 2. Necessity of synchronisation causes the speed-up never to reach 2, what is typical in parallel execution of various algorithms.
2. For single-core computer (PC1 configuration) parallel simulation gives no acceleration, what is obvious, because two simulation processes run in time-sharing. Pseudo-parallel simulation is slower due to synchronisation necessity.
3. Simulation of topology in version B allows better acceleration to achieve, compared to topology version A. In B configuration the communication process is limited comparing to A, less messages are exchanged and processes work more independently.
4. Irrespective the configuration is parallel or serial, increasing Link Delay gives better simulation efficiency, visible as better simulation time achieved. This arises from the

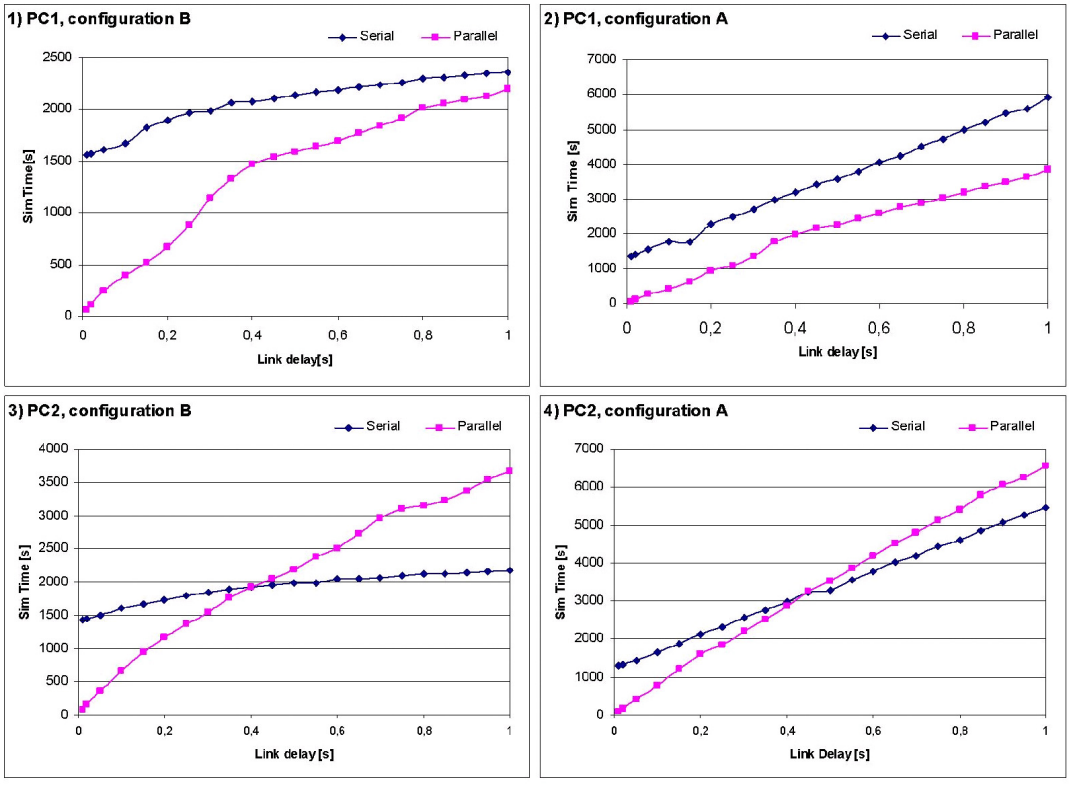



Fig. 2. Dependence between simulation time elapsed and link delay for various configurations

fact, that greater delay effects in less number of messages (segments, packets, frames) per simulation time unit.

Experiments for configuration PC3 (two PC's communicating over MPI) were also performed. However, the results were not reliable, as the participating computers were in fact virtual machines running on host computers. This caused processor power available for simulation process were inconstant and MPI communication delays were big. Experience gained with parallel simulation running with MPI communication will be exploited on cluster machine. Qualitatively, results achieved on PC3 were similar to those on PC2.

## 5. Progress direction – synchronisation algorithms for specific simulation types

As general synchronisation algorithms do not fulfil pinned hopes for significant increase of simulation performance thanks to use of parallel working processors, it seems necessary to look for methods suitable for chosen class of problems. It is typical for parallel processing problems, that no general solution is good enough, and specific problems require specific solutions.

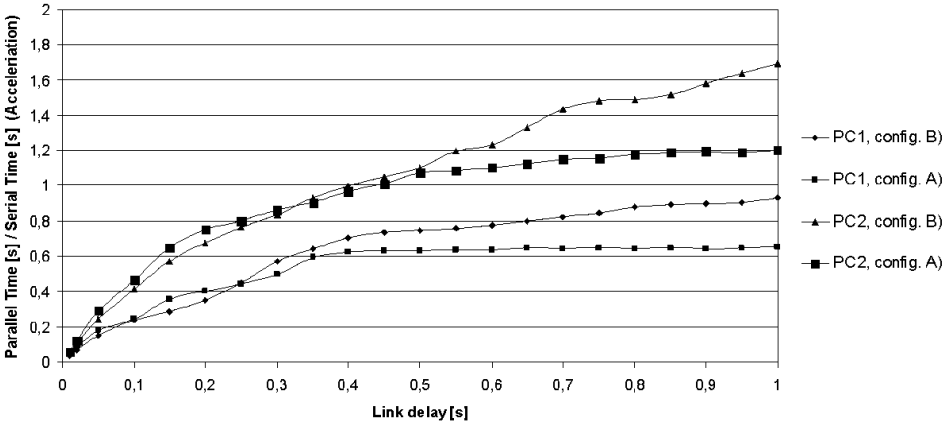



Fig. 3. Simulation acceleration depending of link delay in various configurations

### 5.1. Synchronisation of networks with packet loss

If there was no synchronisation method introduced, LP's would be enforced to drop straggler messages informing on events happened in the past (from the point of view of local simulation time clock). Message dropping would yet lead to simulation errors, as some events would be left without any consequences. In network simulations messages mainly report network packet arrivals. In the case of simulation of a network with packet loss (it refers especially to wireless network, susceptible on electromagnetic noises), loss of some packets is included in the model. Straggler message could be considered as message reporting damaged packet, which content the receiver is unable to read. It is obvious, that mechanisms regulating speed of simulation on particular nodes must be introduced, as the rate of erroneous packets must be possible to set by user. Open problem remains, whether statistical characteristics of a stream of such a packets will be similar enough to real packets streams to consider the simulation results as correct. Described technique, currently worked out by the authors, gives yet hope to elaborate simplified algorithm, possible to use in models with not only non-zero link delays (what is *conditio sin equa non* of every parallel simulation), but also with non-zero rate of damaged packets.

### 5.2. Another mobile wireless networks simulation problems

There is a number of problems specific to wireless networks simulation, for which efficient methods of simulation should be found. Below, most important ones are shortly characterised.

Perhaps the single most difficult and computationally challenging aspect to wireless network simulation is the calculation of the electromagnetic signal strength at a receiver. This signal strength is a function of many variables, including the transmitter power, transmitter

antenna gain factor, receiver antenna gain factor, antenna orientations, terrain obstructions, weather, and ambient electromagnetic interference, just to name a few. Computing signal strength is essential for determining the mobile station will be able to receive the packet or not.

In a wired network environment, the set of directly connected neighbours for a given node is typically constant. However, in wireless environments, nodes almost always are able to move. The simulation environment must take into account station movements and ensure that the simulation produces traffic in a realistic way based on the position of all nodes at any time.

A common model for mobility is the simple *random waypoint* model. In this model, each node randomly chooses an alternate location (called a waypoint) on the geographic region being modelled, and a random velocity. The node is then assumed to move in a straight line from the current location to the new alternate location at the chosen constant speed. When arriving at the new location, the node chooses a random pause time and remains stationary for that amount of time. There exists also variations on the model, like *specific waypoint* model and the *Manhattan* model. ([4])

The effects of mobility, specifically the changing neighbour set for packet transmissions, can be significant for simulation efficiency. For any wireless transmission, any other node in range of the transmitter must be informed of the transmission. In order to be sure that every possible neighbour is informed, the path loss calculations must be performed for every other node in the simulation, to determine if that node is able to receive the signal. Even though at time $t$ a given node $k$ may be unable to detect a transmission from node $n$, there is no guarantee that this is also true at time $t+\Delta t$, since both nodes $k$ and $n$ may have moved during the $\Delta t$ interval. Thus the overall computational complexity of each packet transmission can be as high as $O(N^2)$, where $N$ is the total number of nodes in the simulation. ([4])

Also *ad hoc* routing algorithms, necessary in networks consisting of mobile devices with relatively limited transmission range, can be considerable for both for real networks and for simulators. A showed in [12], in very large networks (50,000 nodes) there can be as many as 500 control packets for every data packet. Clearly, such overhead is a significant computational burden on both the simulator and the actual deployed network. Additionally, the memory requirements for storing routing information in simulator can be $O(N^2)$, where $N$ is the number of nodes in the network.

## 6. Remarks and conclusions

The results presented are only preliminary effects of work over efficient distributed simulator of TCP/IP wireless networks working in cluster environment. However, it is already clear, that for some configuration and parameters use of parallel simulation will be profitable in categories of simulation speed. One can observe it yet on popular dual-core systems. It requires some effort to adapt simulation to parallel system requirements, however for appropriate simulation environment changes will be minor and limited to assigning model to

separate processes (this process can also be automated or at least automatically optimised as shown in [9]).

In succeeding work, except of further adapting OMNeT/INET for work in distributed environments, implementation of new synchronisation algorithm for networks with packed loss is planned. Such an approach limits the set of models which can be simulated, but more efficient simulation is expected.

As it is shown in section 5.2., for effectively exploit problems of wireless network simulation, much research and development is required. Effective solutions for mobile networks should be found and simulation is basic tool to do the research. Therefore, preparing effective and reliable wireless network simulators seems to be nowadays very important task. In IITiS PAN an acquisition of computing cluster with ca. 80 cores is foreseen for the end of this year. It shall be the base for future research on parallel processing problems, especially parallel simulations.

## References


[1] R.E. Bryant. Simulation of packet communication architecture computer systems. *MIT-LCS-TR-188*, 1977.

[2] K.M. Chandy and J. Misra. Distributed simulation: a case study in design and verification of distributed programs. *IEEE Trans. Soft. Eng.*, SE-5(5):440–452, 1978.

[3] R. Fujimoto. *Parallel and Distributed Simulation Systems*. Wiley Series on Parallel and Distributed Computing. John Wiley and Sons, Inc., 2000.

[4] R. Fujimoto, K. Perumalla, and G. Riley. *Network Simulation*, volume Lecture 1 of *SYNTHESIS LECTURES ON COMMUNICATION NETWORKS*. Morgan and Claypool Publishers, 2007.

[5] INET homepage. http://www.omnetpp.org/staticpages/index.php?page=20041019113420757.

[6] D. Jefferson. Distributed simulation and the Time Warp operating system. In *Proc. of the 11th Annual ACM Symposium on Operating System Principles*, pages 77–93, New York, November 1987. ACM Press.

[7] D. Jefferson and H. Sowizral. Fast concurrent simulation using Time-Warp mechanism. In *Distributed Simulation 1985*, pages 63–69, 1985.

[8] A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Inc., 2nd edition, 1991.

[9] M. Nowak. *Dynamiczne Zarządzanie Realizacją Procesów Równoległych w Środowisku Rozproszonym*. PhD thesis, IITiS PAN, Gliwice, 2006.

[10] S. Nowak and J. Domańska. *Sieci komputerowe, tom 2: Aplikacje i zastosowania*, chapter Symulacja protokołu TCP/IP z wykorzystaniem pakietu INET/OMNET++. WKŁ, Warszawa, 2007.

[11] OMNeT++ homepage. http://www.omnetpp.org.

[12] X. Zhang and G. F. Riley. Performance of routing protocols in very large-scale mobile wireless ad hoc networks. In *Symp. on Modeling, Analysis, and Simulation of Computer and Telecommuncation Systems*, Atlanta, GA, USA, 2005.

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 301 -314

# Modeling and Simulation of HTTP Protocol in Networked Control Systems

MAREK FIUK
Pelco
10 Corporate Drive
Orangeburg, NY, USA
*mfiuk@pelco.pl*

**Abstract:** Several modern communication protocols used in networked control and monitoring applications employ combination of XML and SOAP standards, which in turn use the HTTP protocol to provide the actual network communication services. Consequently, we postulate that analysis of the network related performance issues in such applications can be reduced to the performance analysis of the underlying HTTP protocol operating in those environments. In this paper we present a model of a networked control system based on the UPnP communication standard (which is built on top of SOAP, XML and HTTP) together with an efficient algorithm for the analytical examination of that model.

**Keywords:** HTTP, UPnP, modeling, simulation, CTMC, tandem queue

## 1. Introduction

The rapid growth of the Internet created new opportunities for network-based Automation, Control and Monitoring systems. Attracted by the broad range of open standards and protocols (TCP/IP, HTTP, XML. SOAP) numerous teams - from both research and industrial communities, have made attempts to use Internet technologies in Networked Control Systems. The first one was the 1994 Mercury Project at USC which enabled control of a robotic manipulator using the HTTP protocol. It was followed be an Internet-based mobile robot developed at Carnegie-Mellon University in 1995. During the past decade, research teams from around the world have internet-enabled dozens of control applications, including industrial process supervision and control, robotics, home automation, control of telephony switches, operating planetary rovers and spacecraft and many others. In that process several standards and protocols were created, addressing specific needs of particular classes of control and monitoring applications. Included among those are:

- OPC XML DA – OPC Foundation's software interface used in automation industry that adopts the XML set of technologies to facilitate an exchange of plant data across the internet, and upwards into the enterprise domain. It is based on DCOM and Web Services, it follows the Client / Server approach.
- BACnet/WS – an addition to a communication protocol (BACnet) commonly used in the Building Automation Control industry, provides a set of generic Web

Services that can be used to implement an interface to any other building automation protocol.

- VoiceXML – a markup language derived from XML for writing telephone call handling applications. It supports call control, speech synthesis control, and control of voice recognition capabilities through grammars.
- UPnP (Universal Plug and Play) – an architecture and protocols for peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs
- Web Services – set of interfaces describing operations (services) that are network-accessible through the standardized XML messaging. Interfaces are defined using a formal XML notation (service description) that provides details necessary to interact with the service, including message structure, transport protocols, and location.

More often than not these protocols employ combination of XML and SOAP to define necessary data structure and implement the messaging scheme. In turn, these standards use the HTTP protocol to provide the actual network communication services.

Thus, a common pattern can be observed in the architecture of several new communication protocols and standards, used in networked control and monitoring applications (see Fig. 1). The high(er) level control / communication protocol (e.g. BACnet/WS, VoiceXML, UPnP) resides on top of the XML / SOAP combination which in turn resides on top of HTTP. As a result of that, all the networked control applications build around such high level protocols use HTTP as the only network communication platform. Consequently, **analysis of the network related performance issues in networked control applications can be reduced to the performance analysis of the HTTP protocol** used in such applications.

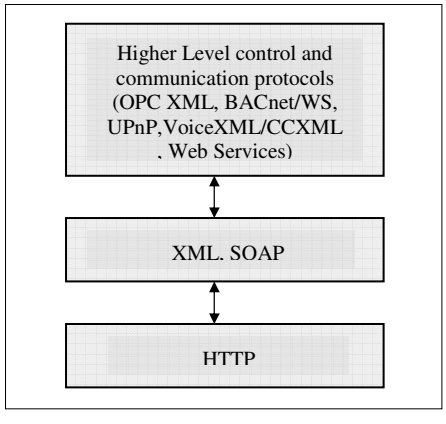



Fig. 1. Typical stack of modern communication protocols

Several different techniques can be employed to perform that analysis. One is to collect and examine performance measurements of a real networked control system, another is to develop analytical apparatus to capture such system's fundamental

properties, yet another is to construct a simulation model and then use it to perform simulation runs to collect performance data.

We have employed both the analytical method and the simulation to study performance of the HTTP protocol in control applications. Furthermore, we have concentrated on systems using UPnP as the higher level control / communication protocol. To that end, we have constructed a model of a networked control system employing combination of UPnP, SOAP, XML and HTTP. In this paper we discuss an efficient algorithm we have developed for the analytical examination of that model. Currently, we are also implementing a simulator that we are going to use to validate the approach taken in the algorithm.

## 2. UPnP - example of an HTTP based control protocol

### 2.1. Protocol overview

Universal Plug and Play (UPnP) is an architecture for peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs. It leverages TCP/IP and the Web technologies (IP, TCP, UDP, HTTP and XML) to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and public spaces.

UPnP allows automatic discovery and control of services available on the network without user intervention. Devices that act as servers can advertise their services to clients. Clients, known as Control Points, can search for specific services on the network. When they find the Devices with the desired services, the Control Points can retrieve detailed descriptions of these services and interact with them from that point on.

UPnP operations can be divided into five basic phases:

- Discovery. In this first phase, Control Points search for devices and services and Devices multicast announcements of services they offer to control points using the Simple Service Discovery Protocol (SSDP). SSDP uses a variant of HTTP that operates over multicast UDP for broadcasts and another variant of HTTP that operates over unicast UDP for replies. To search for Devices or services on the network, Control Points use the HTTP M-SEARCH command multicast to the address 239.255.255.250:1900 over UDP. Any Device on the network that matches the criteria the Control Point is searching for issues a unicast UDP reply that includes the URL to its description document. Devices don't have to wait for a control point to search for their services. They can advertise their device availability by means of the SSDP NOTIFY command on the 239.255.255.250:1900 multicast address.
- Description. Once a Control Point finds an interesting service, it requests from the corresponding Device its complete description. The description is an XML document, it contains (among others) manufacturer information, version, and a list of services supported by the Device. The Control Point requests the description

document using HTTP over TCP. The Control Point performs a standard HTTP GET command (similar to retrieving a Web page).

- Control. This phase allows Control Points to control one or more of the services contained in a Device by enacting changes in the state of the device. The Simple Object Access Protocol (SOAP) allows a control point to query or change elements in a service's state Tab.. SOAP uses the POST or M-POST HTTP command transported over TCP. SOAP uses XML to specify what actions to take. The Control Point creates the XML document and posts it to the control URL for the service, as specified in the description document. The Control Point can request current values and make changes to the service's state Tab..
- Eventing. This phase allows Control Points to keep in sync with the state of services in which it is interested. Control Points subscribe to the event server for a particular service and receive event notifications when that service's state changes. An event notification is sent to the Control Point any time the state of the service changes, even if the Control Point causes the change. Subscribe and unsubscribe requests use HTTP/TCP to connect to the event URL contained in the description document for the service. The Control Point specifies an URL where event notifications are made during subscription. Events arrive by means of HTTP/TCP to the URL registered with the service. The event notification includes a small XML document that describes the actual event, such as a change in the state Tab. for the service.
- Presentation. The presentation phase allows a Device to host a document, written in standard HTML, which can implement a user interface for that device. This document can be downloaded by Control Points and used to perform UI operations, since it provides means of control and status display. The protocol for retrieving the presentation document, as with the description document, is HTTP over TCP. The Control Point can use the presentation URL contained in the description document to request the presentation document.

## 3. UPnP network model

We present here a model of an UPnP network operating in the control application environment. It was constructed based on the UPnP specification document "Universal Plug and Play Device Architecture" and an analysis of the source code provided in the "Intel® SDK for UPnP™ Devices Version 1.2.1" package.

Two types of UPnP network operations can be distinguished - the "Power-up/ Network Initialization" operations and the "Control Phase" operations. Since Control Points and Devices can be powered-up / attached to or detached from the network at any time, operation of both types can potentially overlap on the network. For example, a Control Point can receive SSDP NOTIFY command from a Device while sending SOAP request (Control Phase) to another Device. However, under typical conditions a Device will be involved in the Control Phase operations much more often than in the Power-up Phase operations (will send and receive many more Control Phase Commands and Event

Notifications). Therefore, currently only the Control Phase operations are reflected in the model.

Fig. 2 presents the UPnP Queuing Network (QN) Model for the Control Phase Operations. It consists of Control Point elements (numbered 1 – P), Device elements (numbered 1 – Q) and a single Network element. Control Points and Devices are all connected to the Network with connections symbolizing either the UPnP Request flow (solid line) or the UPnP Response flow (dashed line).

### 3.1. Control Phase operations

A UPnP Control Point generates Control Requests at the rate $G_r$ (requests per second) each destined to a single UPnP Device. Requests generated by a given Control Point may all go to the same Device or may be distributed among several Devices. After a Control Request is sent to its target Device, the Control Point continues to keep track of its status until that Device indicates that the request was received and fully processed (consumed). The Control Point is capable of maintaining (keeping track of) K outstanding Control Requests.

Once received, the Control Requests are serviced by the Device, with each request requiring processing time $t_{pc}$. The Device is capable of processing M request simultaneously. Control Requests may be used to affect the Plant (controlled object) associated with the Device by changing value of its State Variables, or to query the status of that Plant. Changes to State Variables may in turn result in sending the Event Notification Request to all Control Points that registered for that particular event. There exist a correlation factor (coefficient) $M_s$ between the number of received Control Requests and the number of resulting *State Variable Change* Event Notification Requests.

After processing is finished, the Device generates the Control Response which is sent back to the Control Point thus completing the handling of the Control Request there.

As the Plant's state changes (due to the processing of the Control Requests and the influence of the external environment) it generates (at the rate $G_e$) *Plant State Change* Event Notification Requests that are sent to Control Points that registered for them. After the Event Notification Request is sent to a Control Point, the Device continues to keep track of its status until that Control Point indicates that the request was received and fully processed (consumed).

Once received, the Event Notification Requests are processed by the Control Point, with each Notification Request requiring processing time $t_{pe}$. The Control Point is capable of processing T Event Notification Requests simultaneously. After processing is finished, the Control Point generates the Event Notification Response which is sent back to the Device thus completing the handling of the corresponding Event Notification Request there.

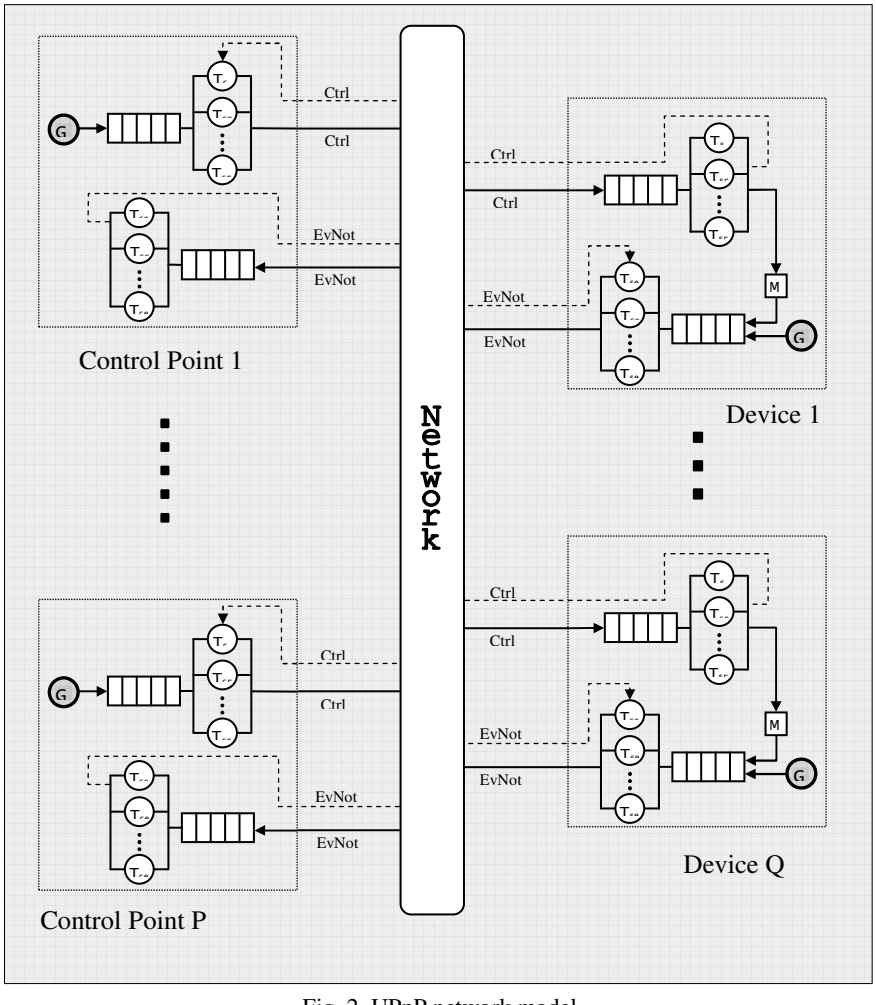



Fig. 2. UPnP network model

### 3.1.1. The Control Point

Control Requests generated by the source $G_r$ are queued before being sent to Devices by the concurrent transmission tasks $T_{cr1}$ - $T_{crK.}$ After sending a request, the transmission task waits for the Control Response to arrive from the Device and only then it can proceed to process next request from the queue. Event Notification requests received from Devices are queued before being processed by the concurrent event handling tasks $T_{ce1}$ - $T_{ceT}$ (this processing includes sending an Event Notification Response back to the Device).

### 3.1.2. The Device

Control Requests received from Control Points are queued before being processed by the concurrent tasks $T_{sr1}$ - $T_{srM}$. For each processed Control Request a corresponding Event Notification Request will be generated (with a probability of $M_s$) and queued. After processing of the request is completed, the task sends a Control Response back the Control Point.

Event Notification Requests (for the events corresponding to operations of the Plant) generated by the source $G_e$ are queued before being sent to a Device by the concurrent transmission tasks $T_{se1}$ - $T_{seN.}$ After sending a request, the transmission task waits for the Event Notification Response to arrive from the Control Point and only then it can proceed to process next request from the queue.

### 3.1.3. The Network

Currently, a simplistic Constant Delay network model is used.

## 4. UPnP network analysis

The complexity of the UPnP network model presented in the previous paragraph makes its analysis rather difficult. Therefore, its further reduction is needed before formal analysis can be attempted.

It can be easily seen in the network model that any communication requires an HTTP connection between the HTTP sender (consisting of a queue and a number of sending tasks) and the HTTP receiver (consisting of a queue and a number of request/job processing tasks). Although a particular connection can be of one of the two types - the first involving the Control Point sending the Control Requests to the Device ( with sending tasks $T_{cr1}$ - $T_{crK}$ and request processing tasks $T_{sr1}$ - $T_{srM,}$) and the second involving the Device sending the Event Notification Requests to the Control Point (with sending tasks $T_{se1}$ - $T_{seN}$ and request processing tasks $T_{ce1}$ - $T_{ceT}$), the structure of the sender / receiver arrangement is identical in both cases. Since the outlined above HTTP sender / receiver relationship is of the client / server nature, we will use the later terms for the remaining of this paragraph. Thus, we reduce analysis of the UPnP network model to the analysis of an HTTP client / server pair which can be represented by a two-node tandem queue arrangement with an additional blocking mechanism, as shown in Fig. 3.

There are several parameters defined for this new model, among them:

$K_c$ – total number of jobs at the client $K_c = S_c + Q_c$

$K_s$ – total number of jobs at the server $K_s = S_s + Q_s$

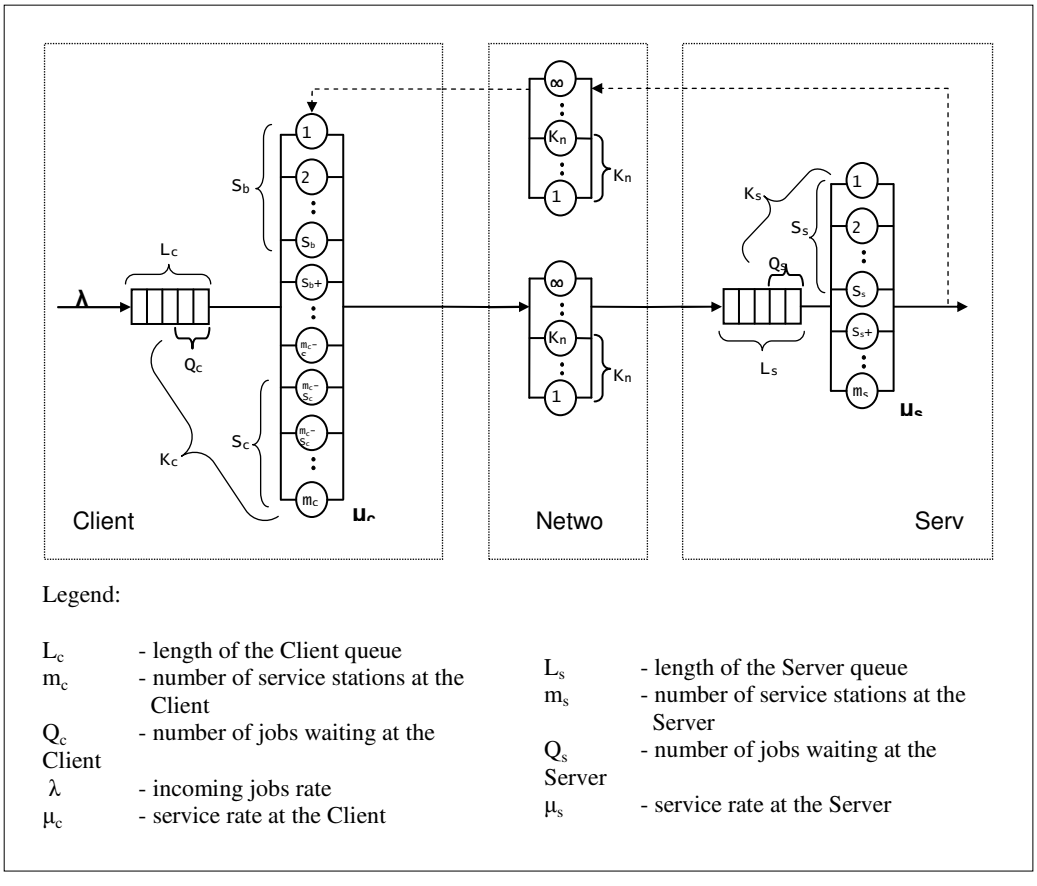
$S_b$ – number of client service stations blocked by jobs waiting/serviced at the server and jobs in the network $S_b = K_s + K_{nf} + K_{nr}$

$S_c$ – number of jobs currently serviced at the client $0 \le S_c + S_b \le m_c$

$S_s$ – number of jobs currently serviced at the server

We limit our considerations to the cases where the time needed to process a job at both the client and the server is significantly larger than time needed to deliver messages over the network, therefore we assume in our analysis that there are no "in flight" jobs in the system, e.g. $K_{nf}$, $K_{nr} = 0$.

The blocking mechanism present in the system makes the ratio between the number of service stations at the client ($m_c$) and the total capacity of the server ($L_s + m_s$) significant. We assume that the system is balanced in respect to that ratio, e.g. $m_c = L_s + m_s$. The rational here is that in a system with $m_c < L_s + m_s$ some of the server resources (queue, service stations) will simply never get fully utilized, therefore such system will effectively get reduced to a balanced one with smaller $L_s$ and (possibly) $m_s$.




Legend:

| | | | |
|---|---|---|---|
| $L_c$ | - length of the Client queue | $L_s$ | - length of the Server queue |
| $m_c$ | - number of service stations at the Client | $m_s$ | - number of service stations at the Server |
| $Q_c$ | - number of jobs waiting at the Client | $Q_s$ | - number of jobs waiting at the Server |
| $\lambda$ | - incoming jobs rate | | |
| $\mu_c$ | - service rate at the Client | $\mu_s$ | - service rate at the Server |

Fig. 3. Tandem queue model of the HTTP sender / receiver pair

On the other hand, in a system with $m_c > L_s + m_s$ some of the jobs leaving the client may get dropped, but since the space in the servers queue is usually inexpensive, it is unlikely that such (possibly harmful) imbalance would be allowed in a practical application. Therefore, we consider this case highly artificial.

We use Continuous Time Markov Chains (CTMC) to analyze the two-node tandem queue system that represents the reduced UPnP network model. Such a tandem queue system can be described by a phase/level random process with the phase ($K_c$ in our case) giving the state of the client and the level ($K_s$ in our case) giving the state of the server. The combination of the phase and the level defines the state of the whole system. The transitions between states can be represented by the state flow diagram which in turn is used to construct a state transition matrix **Q**.

Assuming that incoming jobs have Poisson distribution with arrival rate $\lambda$, and assuming that client and server service times also have Poisson distribution with service rates respectively $\mu_c$ and $\mu_s$, we can apply the CTMC theory to construct the set of global balance equations:

$$\mathbf{\Pi} * \mathbf{Q} = \mathbf{0}$$

where **Π** is a vector of steady state probabilities of system being in particular state (e.g. having a particular combination of phase and level values). Together with the normalization equation:

$$\mathbf{\Pi} * \mathbf{1} = 1$$

these equations can be solved yielding the vector **Π**, which fully describes stochastic properties of the analyzed tandem queue system and thus properties of the reduced UPnP network model.

As a concrete illustration of our approach, Fig. 4 presents the state flow diagram and Fig. 5 presents the state transition matrix that correspond to a very small tandem queue system (UPnP network model) with following parameters:

$$L_c = 1,\ m_c = 3,\ L_s = 1,\ m_s = 2$$

The global balance equations can be solved directly using Gaussian Elimination. This method consists of two steps – matrix triangularization with arithmetic complexity of $n^3/3$ and the backward substitution with arithmetic complexity of $n^2/2$, where n is the size of the matrix **Q** which is equal to the number of states in the analyzed tandem queue system.

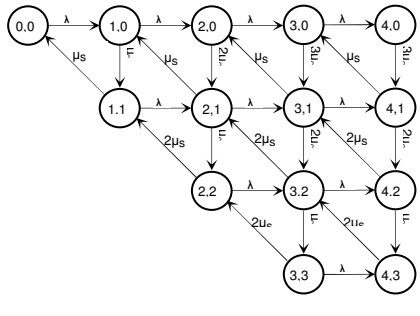



Fig. 4. State flow diagram.

The number of arithmetic operations needed for a solution algorithm that has arithmetic complexity of $n^3/3$ grows rapidly with n. This makes limiting this complexity a very desirable goal. To that end, we have employed two methods – one is to reduce the

number of elements that have to be eliminated in the matrix triangularization process, second is to avoid fetching and processing zero valued elements of **Q**.

```
-λ,    0,   us,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0;
 λ,-uc-λ,    0,    0,   us,    0,    0,    0,    0,    0,    0,    0,    0,    0;
 0,   uc,-us-λ,    0,    0, 2*us,    0,    0,    0,    0,    0,    0,    0,    0;
 0,    λ,    0,-2*uc-λ,  0,    0,    0,   us,    0,    0,    0,    0,    0,    0;
 0,    0,    λ, 2*uc,-us-uc-λ, 0,    0,    0, 2*us,    0,    0,    0,    0,    0;
 0,    0,    0,    0,   uc,-2*us-λ,  0,    0,    0, 2*us,    0,    0,    0,    0;
 0,    0,    0,    λ,    0,    0,-3*uc-λ,  0,    0,    0,    0,   us,    0,    0;
 0,    0,    0,    0,    λ,    0, 3*uc,-2*uc-us-λ,0,   0,    0,    0, 2*us,    0;
 0,    0,    0,    0,    0,    λ,    0, 2*uc,-2*us-uc-λ,0,   0,    0,    0, 2*us;
 0,    0,    0,    0,    0,    0,    0,    0,   uc,-2*us-λ,  0,    0,    0,    0;
 0,    0,    0,    0,    0,    0,    λ,    0,    0,    0,-3*uc,    0,    0,    0;
 0,    0,    0,    0,    0,    0,    0,    λ,    0,    0, 3*uc,-2*uc-us, 0,    0;
 0,    0,    0,    0,    0,    0,    0,    0,    λ,    0,    0, 2*uc,-2*us-uc, 0;
 0,    0,    0,    0,    0,    0,    0,    0,    0,    λ,    0,    0,   uc,-2*us
```

Fig. 5. State transition matrix

The first method rearranges the set of global balance equations to eliminate the leftmost non-zero element in each row of matrix **Q** thus producing new (transformed) matrix **Q''**. This is accomplished by (conceptually) adding to each row the row that precedes it, starting with the second row and advancing to the last one, and then, for each phase $r \leq m_c$, restoring the last rows in that phase to its original form. In other words, given matrix **Q** with rows $\mathbf{R}_0$ - $\mathbf{R}_{n-1}$:

$\mathbf{R'}_0 = \mathbf{R}_0$; // e.g. unchanged

for(k=1; k<n; k++)

$\mathbf{R'}_k = \mathbf{R}_k + \mathbf{R'}_{k-1}$;

$\mathbf{R''}_0 = \mathbf{R'}_0$;

for(k=1; k<n; k++)

if(phase $\leq m_c$ and row k is the last one in the phase)

$\mathbf{R''}_k = \mathbf{R'}_k - \mathbf{R'}_{k-1}$;

else $\mathbf{R''}_k = \mathbf{R'}_k$;

Given that from the definition of matrix **Q**, for every column j ($0 \leq j < n$) we have:

$$\sum_{k=0}^{n-1} q_{k,j} = 0$$

and since the leftmost non-zero element in a row is the last one in its column, the above algorithm indeed eliminates this element. Fig. 6 presents such a transformed matrix **Q''** for the original matrix **Q** from Fig. 5.

It needs to be noted here that this is a conceptual transformation only – it doesn't need to be actually performed, since the structure of the transformed matrix **Q''** is rather clear, and it can be constructed directly, given l,uc,us and n !

The second method of reducing the matrix **Q''** triangularization complexity is to avoid fetching and processing its zero valued elements. Owing to the structural regularity of the phase/level process' state flow diagram, the structure of the matrix **Q''** exhibits strong patterns. The pattern of interest here is the number of elements preceding and following the diagonal one in a particular row, expressed as a function of the level this row belongs to and of the position of the row within that level. Tab. 1 summarizes these regularities.

```
-λ,    0,   us,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0;
 0,-uc-λ,   us,    0,   us,    0,    0,    0,    0,    0,    0,    0,    0,    0;
 0,   uc,-λ-us,    0,    0, 2*us,    0,    0,    0,    0,    0,    0,    0,    0;
 0,    0,   -λ,-2*uc-λ, us, 2*us,    0,   us,    0,    0,    0,    0,    0,    0;
 0,    0,    0,   -λ,-uc-λ, 2*us,    0,   us, 2*us,    0,    0,    0,    0,    0;
 0,    0,    0,    0,   uc,-λ-2*us,  0,    0,    0, 2*us,    0,    0,    0,    0;
 0,    0,    0,    0,   -λ,   -λ,-3*uc-λ, us, 2*us, 2*us,    0,   us,    0,    0;
 0,    0,    0,    0,    0,   -λ,   -λ,-2*uc-λ,2*us,2*us,    0,   us, 2*us,    0;
 0,    0,    0,    0,    0,    0,   -λ,   -λ,-uc-λ, 2*us,    0,   us, 2*us, 2*us;
 0,    0,    0,    0,    0,    0,    0,    0,   uc,-λ-2*us,  0,    0,    0,    0;
 0,    0,    0,    0,    0,    0,    0,   -λ,   -λ,   -λ,-3*uc,   us, 2*us, 2*us;
 0,    0,    0,    0,    0,    0,    0,    0,   -λ,   -λ,   -λ,    0,-2*uc, 2*us, 2*us;
 0,    0,    0,    0,    0,    0,    0,    0,    0,   -λ,   -λ,    0,    0,  -uc, 2*us;
 0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0;
```

Fig. 6. Transformed state transition matrix

| Phase (r) | Row within phase (n) | Number of preceding elements (p) | Number of following elements (f) |
|---|---|---|---|
| r = 1 ÷ $m_c$-1 | n = 0 ÷ r-1 | r - 1 | r + 2 |
| | n = r | 1 | r + 2 |
| r =$m_c$ | n = 0 ÷ r-1 | r - 1 | r + 2 |
| | n = r | 1 | r + 1 |
| r = $m_c$+1 ÷ $m_c$+$L_c$-1 | n = 0 ÷ $m_c$-1 | $m_c$ | $m_c$+2 |
| | n = $m_c$ | 1 | $m_c$+1 |
| r =$L_c$ | n = 0 ÷ $m_c$-1 | $m_c$ | $m_c$-n |
| | n = $m_c$ | $m_c$ (0) | 0 |

Tab. 1. Regularities in the state transition matrix

Utilizing this regularities, we have constructed an algorithm for the triangularization of a (transformed) matrix **Q''.** A simplified C implementation of this algorithm is shown in Fig.s 7 and 8. It operates on a global matrix mQ, which needs to be initialized with **Q''.** To execute the procedure function `gElimFRA()` needs to be invoke with `mc` – number of service stations at the Client, and `Lc` – length of the Client queue, passed as

parameters. The processing (triangularization) of the entire matrix is subdivided into processing of sets of rows corresponding to an individual phase, which is performed by function `phsElim()`. As shown in Tab. 1, four different types of phases can be distinguished - phases $r = 1 \div m_c-1$, phase $r = m_c$, phases $r = m_c+1 \div m_c+L_c-1$ (if at all present), and phase $r = L_c$, where r is equal to the total number of jobs at the client (e.g. $K_c$).

```
gElimFRA(mc,Lc)
{
N = (mc+1)*((mc+2)/2 + Lc);
elim1El(N-1,0,3);
fr = 1;
for(r=1;r<mc;r++) {
   phsElim(fr,r+1,r-1,1,r+2,r+2,r+1,r+1,r,N);
   fr += r + 1;
   }
phsElim(fr,mc+1,mc-1,1,mc+2,mc+1,mc+1,mc+1,mc,N);
fr += mc + 1;
if(Lc > 1)
   for(k=0;k<(Lc-1);k++) {
      phsElim(fr,mc+1,mc,mc,mc+2,mc+1,mc+2,mc+1,mc+1,N);
      fr += mc + 1;
      }
phsElim(fr,mc,mc,mc,-mc,0,mc+2,mc+1,mc+1,N);
}
```

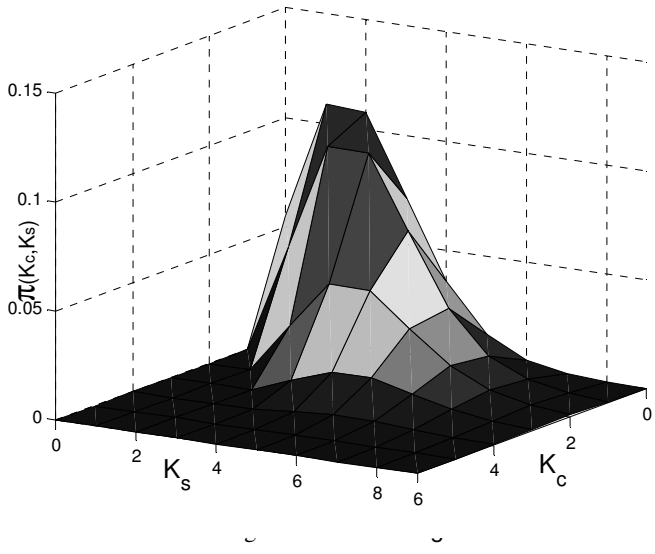
Fig. 7. Matrix triangularization algorithm

```
phsElim(lf,h,p,pl,f,fl,pf,pfl,ph,ms)
{
if(f < 0) {
   f = -f;
   dlt = 1;
   } else dlt = 0;
for(i=0;i<h;i++) {
   cr = lf + i;
   if(i == (h - 1))
      if(dlt == 1) {
         ap = p;
         af = f - dlt * i;
         }
      else {ap = pl; af = fl; }
   else { ap = p; af = f - dlt * i; }
   for(j=0;j<ap;j++) {
      cur = cr - ap + j;
      if(cur < lf)
         if(cur == (lf - 1)) cnt = pfl + 1;
         else  cnt = pf + 1;
      else cnt = f - dlt * (cur - lf) + 1;
      elim1El(cr,cur,cnt);
      }
   elim1El(ms-1,cr,af+1);
   }
}

elim1El(p,a,cnt)
{
s = mQ[p,a] / mQ[a,a];
for(nn=a;nn<(a+cnt);nn++)
   mQ[p,nn] -= mQ[a,nn]*s;
}
```

Fig. 8. Matrix triangularization algorithm (cont.)

Using the above algorithm followed by the standard backward substitution procedure, we can efficiently compute values of the steady state probabilities vector **Π.** Fig. 9 shows such values plotted as a function of the phase and the level e.g. $\mathbf{\Pi}(K_c, K_s)$ .




**4.1. Complexity analysis**

Total complexity of the triangularization algorithm presented here can be expressed as:

$S = (L_c - 1)(m_c + 1)( m_c (m_c + 3) + (m_c + 2)) + (3m_c^4 + 26m_c^3 + 69m_c^2 + 106m_c + 24) / 12$

It is customary to express complexity of triangularization algorithms as a function of matrix size. Given that the size N of the state transition matrix **Q''** can be expressed as:

$N = (m_c + 1)((m_c + 2) / 2 + L_c)$

it can be easily shown that $S < N^2$ for large $m_c$ and $L_c$. Thus the computational complexity of the triangularization algorithm presented here is less than $O(n^2)$.

## 5. UPnP network simulation

In order to construct an efficient algorithm for the analytical examination of the UPnP network model we have reduced analysis of that model to the analysis of an HTTP client / server pair which we represent by a two-node tandem queue arrangement. To validate this approach

we are currently implementing a UPnP network simulator which we will use to generate data that can be compared with the results produced by our optimized CTMC solution algorithm. It directly implements (simulates) the UPnP Queuing Network Model for the

Control Phase Operations, as presented in Fig. 2. Simulator employs the discrete event simulation technique; it was constructed using the OMNeT++ simulation framework.

### 6. Conclusions and further work

Several modern communication protocols, used in networked control and monitoring applications employ combination of XML and SOAP standards, which in turn use the HTTP protocol to provide the actual network communication services. Consequently, we postulate that analysis of the network related performance issues in such applications can be reduced to the performance analysis of the underlying HTTP protocol operating in those environments.

To that end, we have constructed a model of a networked control system employing combination of UPnP, SOAP, XML and HTTP. We then developed an efficient algorithm for the analytical examination of that model. Currently, we are building a network simulator to validate the results generated by our algorithm.

## References

[1] Universal Plug and Play Device Architecture, Version 1.0, 1999-2000 Microsoft Corporation.

[2] Linux SDK for UPnP Devices Version 1.2, 2000-2003 Intel Corporation.

[3] Czachórski T., Modele kolejkowe w ocenie efektywnosci sieci i systemów komputerowych, Pracownia Komputerowa Jacka Skalmierskiego, Gliwice 1999.

[4] OPC Foundation,OPC XML-DA Specification,Version 1.0, July 2003.

[5] Proposed Addendum C to Standard 135-2004, BACnet - A Data Communication Protocol for Building Automation and Control Networks, ASHARE 2004.

[6].Menasce D., Almeida V., Capacity Planning for Web Services, Prentice-Hall, 2002

Polish Teletraffic Symposium 2007
ISBN 978-83-926054-0-9
pp. 315–323

# Simulation Analysis of Deflection Routing in Hypercube

KRZYSZTOF GROCHLA[1] TADEUSZ CZACHÓRSKI[1] ANNA BUSIC[3]
JEAN-MICHEL FOURNEAU[2,3]

[1] IITIS-PAN, Ul. Bałtycka 5, 44-100 Gliwice, Poland
[2] INRIA project MESCAL, Grenoble, France
[3] PRiSM, Université de Versailles Saint-Quentin,
45 Av. des Etats Unis, 78000 Versailles, France

**Abstract:** The article gives some performance evaluation results for the all-optical networks with hypercube topology. It contains description of the simulation model, performance evaluation results from simulation and two theorems verified by the simulation. In the 3rd section the plots for link loads and deflection probability under different load are given. Then the equation for deflection probability and average end-to-end delay is given and compared to the simulation - what gives some cross-validation of both models.
**Keywords:** Hypercube, deflection routing, all-optical networks

## 1. Introduction

Electrical to optical conversion becomes now a bottleneck of the fast computer networks. Thus new network architectures are proposed, without the need of transmitted data conversion within the core nodes. Among a few others, the all-optical packet switched network technology was proposed. It is based on a constant size packets with small header send in constant time intervals, called slots. As there is no cheap optical memory on the market, the packet can not be stored in the optical switches and the traditional "store and forward" algorithms, used in nowadays networks, can not be used. The decision, where a packet should be sent have to be done in a very short period of time, so computing the transmission path cannot be very complex.

Old algorithms like Deflection Routing [1] have recently received attention to overcome this weakness [3], [4]. This routing strategy does not allow packet loss but it keeps the packets inside the network, increases the delay and reduces the bandwidth. In Shortest-Path Deflection Routing, switches attempt to forward packets along a shortest hop path to their destination. Each link can send a finite number of packets per time-slot: this is the link capacity. If the number of packets which require a link is larger than the capacity, only some

of them will use the link and the others have to be misdirected or deflected and they will travel through longer paths.

In this work we consider the hypercube topology, which is not common in telecommunication networks, but used in high speed interconnect networks in supercomputers or clusters. In the hypercube of size $n$ all nodes have $n$ links. The maximum distance in hops is equal to $n$.

## 2. Model assumptions

### 2.1. Hypercube

We model a hypercube of dimension $n$ (see figure 1 for a hypercube of dimension 4). A hypercube is a simple generalization of a cube with an arbitrary size. The nodes' addresses can be represented as vectors with $n$ components taking values in 0, 1. Nodes which differ only by one component are connected by two directed edges. Thus a hypercube of dimension $n$ has $2^n$ nodes and $n2^n$ directed edges. Switches have thus an indegree and outdegree equal to $n$. The shortest path distance in the hypercube is equal to Hamming distance among binary vectors. The diameter of a hypercube with dimension $n$ is thus $n$.

### 2.2. Routing algorithm

Let X and Y be two nodes. All the directions such as $X(i) \neq Y(i)$ are good directions for the routing algorithm to send a packet from $X$ to $Y$ using shortest paths. All the others are bad directions. Therefore a packet at distance $k$ of its destination has $k$ good directions for the next step of routing. In figure 1 we give an example of a packet with current position 1011 and destination 0101. Thus the distance to destination of the packet is 3 and it has 3 good directions (depicted in bold): 0011, 1111 and 1001.

The packet will select at random with uniform distribution one direction among the good directions. If this direction is not given to the packet by the routing algorithm, the packet is deflected. We say that this direction is not available. We consider a two phases algorithm instead of a greedy choice. During the first phase the packets which are not deflected are routed and the deflected packets are kept. Then during the second phase the deflected packets are sent among the directions which are still available after the first phase. A deflected packet uses a direction at random with uniform distribution among all available directions.

### 2.3. Simulation model

We assume that there is no memory in the switches, what makes it difficult to apply classical queuing theory to analyze the delay. Thus a simulation model have been created in OMNeT++[5] environment. The model was used to analyze the performance measures of the deflection routing in hypercube and to verify some mathematical equations describing the network.

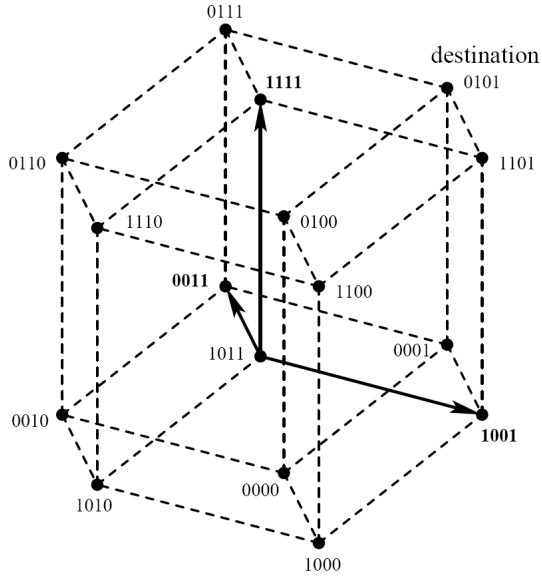



Fig. 1. The hypercube topology

In the simulation model all nodes are represented by compound modules in OMNeT++. Each of the nodes can generate and consume traffic. The node consists of three submodules: *generator*, *switch* and *sink*. The routing algorithm is implemented in the *switch* class. The structure of the model can be seen on Fig. 2. The packets are represented by messages in OMNeT++ - for each packet the travel time and number of deflections is stored. The data are collected and summarized in the *sink* module. The simulation time is calculated in time slots. In the results presented in the paper the exponential generators were used, however the simulator supports any of the statistical generators available in OMNeT++[5].

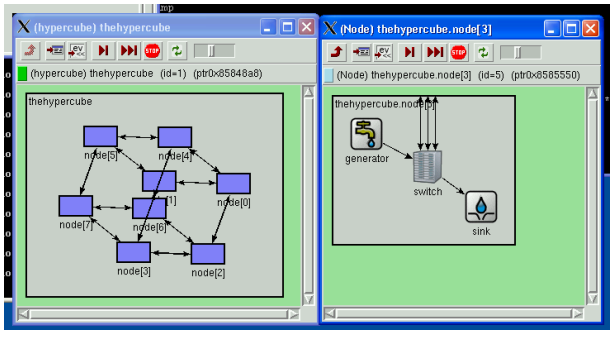



Fig. 2. Structure of the simulation model

## 3. Simulation results

The simulation model was used to evaluate the performance of hypercube of size 7. The packets were generated uniformly in all network nodes using exponential generators. The simulations were run for mean intergeneration time from 1 to 3.5 slots to create different network load. The global link load (probability, that in arbitrary time a link in a network is used), deflection probability (probability, that a packet will reach the destination without any deflection) and switch load (number of links used in a switch divided by number of links in a switch) are presented on Fig. 3 - Fig. 6

The Fig. 6 presents the mean packet travel time for different network load given by the simulation. The network was uniformly loaded. The mean travel time for empty network is equal to 4 (the average distance in the network). The result shows that the deflection routing algorithm stays stable in the hypercube even when the load is greater then 0.7 - the average travel time is less then 2 times the minimum value.

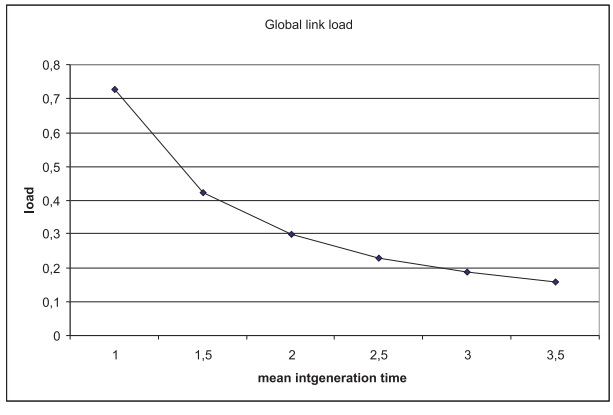



Fig. 3. Link load for different intergeneration time

## 4. Verification by simulation some assumption used in mathematical modeling of hypercube

The next task was to use the simulation to verify some of the assumptions made in creation of the mathematical model of deflection routing in hypercube. The detailed model is given in [2], below some basics of equations evaluated are given.

### 4.1. Model assumptions

We represent a arbitrary packet by its distance to destination. The random variable $X$ is between $0$ and $n$. We do not need type of packets. Let $p$ be the deflection probability. $p$ will be derived in the next section.

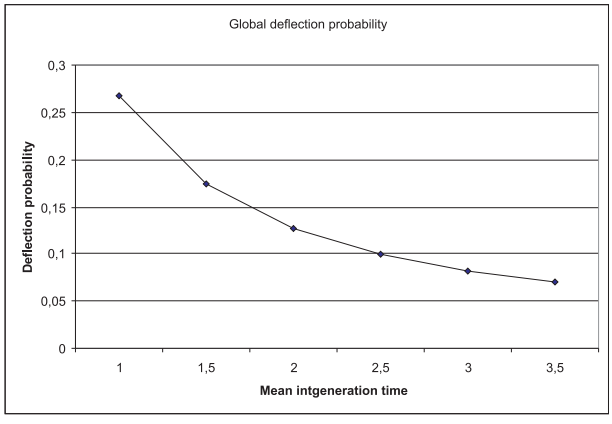



Fig. 4. Packet deflection probability for different intergeneration time

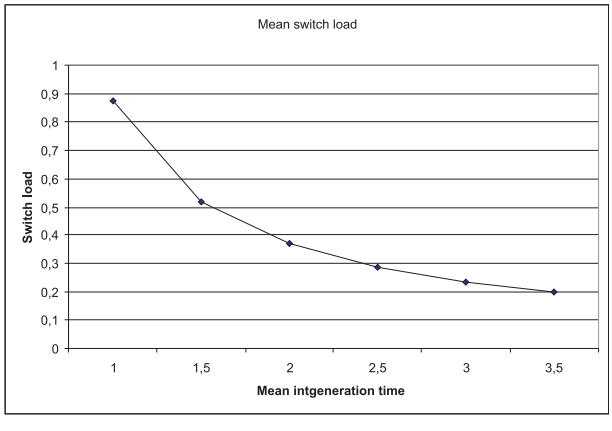



Fig. 5. Switch load for different intergeneration time

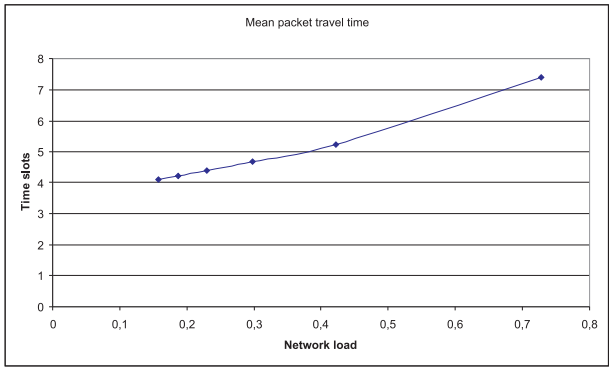



Fig. 6. Travel time for different network load

- Assume that the packet at distance $k$ is not deflected, the distance decreases from $k$ to $k-1$. This event has probability $(1-p)$.
- If the packet at distance $k$ is deflected, it remains $(k-1)$ good directions among $(n-1)$ available directions. If the packet uses a good direction, its distance is now $(k-1)$; otherwise it is $(k+1)$. Thus we have the following transitions:
  - $k$ to $k-1$ with probability $p\frac{k-1}{n-1}$
  - $k$ to $k+1$ with probability $p\frac{n-k}{n-1}$

  When $k=1$, the deflected packet always uses a bad direction. When $k=n$, whenever the packet is deflected or not, it uses a good direction.

### 4.2. Deflection probability

As all the packets are equivalent in a probabilistic point of view, we consider an arbitrary packet in an arbitrary switch. Note that due to topology and traffic assumptions all the switches are statistically equivalent. The probability of deflection for a tagged packet is computed by conditioning on all configurations of arrivals. Note that the upper bound of the index is $n-1$ because the tagged customer uses one input link of the switch.

$$p = \sum_{i=0}^{n-1} Pr(i\ other\ packets\ destinated\ to\ link) Pr(deflection\ of\ a\ tagged\ packet \mid i) \tag{1}$$

The arrival probabilities are obtained by an independence assumption. Let us denote by $u$ the utilization of an arbitrary link. We have $n-1$ links. Each of them is used by a packet with probability $u$. When a packet enters a switch, it requires each output link with probability $1/n$.

$$Pr(i\ other\ packets\ destinated\ to\ the\ output\ link) = C(n-1,i)(u/n)^i(1-u/n)^{n-1-i} \tag{2}$$

The probabilities of deflection can be calculated as follows. The output capacity is 1. We have $i+1$ packets in a fair competition. Thus the tag packet wins with probability $1/(i+1)$ and is deflected with probability $i/(i+1)$.

$$Pr(deflection\ of\ a\ tagged\ packet \mid i) = \frac{i}{i+1} \tag{3}$$

The equation 3 was verified using the simulator. The plots Fig. 7 and Fig. 8 presents the deflection probabilities for packets experiencing given number of deflections for two sample network loads. The plots compare the results given by the equation 3 and simulation. As one can see the results match almost perfectly. The only problem relates to the packets experiencing very high number of deflections (6 in this case), as there are very rare - only a few packets in the simulation of 1000000 time slots, with tens of millions of packets experiencing one or no deflections.

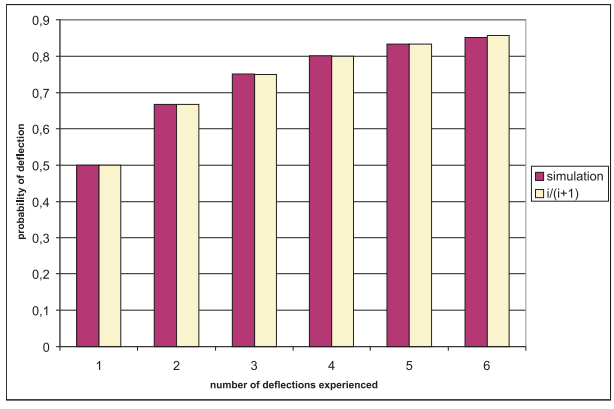



Fig. 7. Deflection probability for ig time 1 for packets experiencing different number of deflections

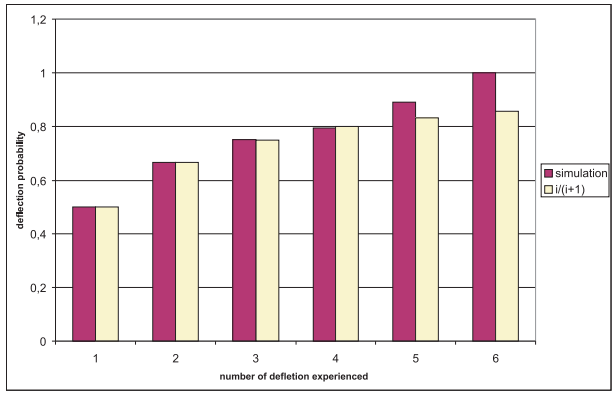



Fig. 8. Deflection probability for ig time 2 for packets experiencing different number of deflections

### 4.3. Average End to end Delay

Let us now establish a new relation between the link utilization $u$ and the deflection probability $p$.

Let $E(X)$ the expected number of customers in the network, $\lambda$ the input rate in the global set of nodes from the electronic buffers and $E(T)$ the average end to end delay. $E(T)$ is the average number of hops (i.e. the average sojourn time in the optical part of the network) Following Little's law we get:

$$E(X) = \lambda E(T)$$

All the links are equivalent due to the topology and the traffic assumptions, we have $2^n$ nodes and each nodes has $n$ input links. Thus:

$$u = \frac{E(X)}{n2^n}$$

If the system is stable the input rate in the optical part $\lambda$ is equal to the input rate in each electronic buffer $\gamma$ multiplied by the number of nodes $2^n$. We also have $u \leq 1$. Finally:

$$u = min(1, \frac{\lambda E(T)}{n2^n}) \tag{4}$$

This equation was verified using the simulator. The Fig. 9 presents comparison of $E(t)$ and $E(x)$ from a simulation of the network of dimension $n = 7$. The results almost perfectly match each other, what cross-validates the simulation model and above theorem.

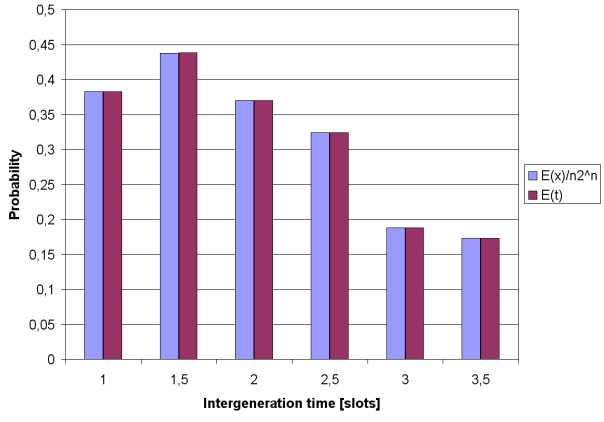



Fig. 9. Deflection probability for different network load by simulation and eq. 4

## 5. Summary

The all-optical networks are promising technology for future very fast networks. The hypercube topology is very effective e.g. to cluster nodes interconnection and may be effectively used in many small-distance networks. Thus detailed analysis of this technologies is required. The authors give some simulation results for this type of networks, as well as some mathematical analysis verified by the simulation.

## References

[1] P. Baran. "On distributed communication networks" IEEE Transactions on Communication Systems, CS-12:1–9, 1964.

[2] A. Busic, Czachórksi T., J.M. Fourneau, K. Grochla "Level Crossing Ordering of Markov Chains: Proving Convergence and Bounding End to End Delays in an All Optical Network", Proceedings from ICST Valuetools, Nantes, 2007

[3] P. Gravey et al. "Multiservice optical network: Main concepts and first achievements of the ROM program" Journal of Ligthwave Technology, 19:23–31, 2001.

[4] S. Mneimeh and F. Quessette. "Minimum deflection routing algorithm", alcatel patent application no. 135945, 2002.

[5] Varga A. "‘The OMNeT++ Discrete Event Simulation System"’, Proceedings of the European Simulation Multiconference (ESM’2001), Prague 2001.

ISBN 978-83-926054-0-9