Paper

Impact of Signaling System Performance on QoE in Next Generation Networks

Jordi Mongay Batalla, Jarosław Śliwiński, Halina Tarasiuk, and Wojciech Burakowski

Abstract—The first experience of quality by multimedia applications' users takes place during the setup phase of a new connection. If the setup phase is not accepted or "slowly accepted", the confidence of the user decreases. The user becomes more sensitive when he/she pays the connections with assured quality of service (QoS). In this case, the process of call request should be also accomplished with QoS guarantees. This paper presents the signaling sub-system implemented within the EuQoS system. The EuQoS signaling process follows main assumptions of next generation networks (NGN) architecture and performs tasks related with codec agreement between multimedia end users, admission control and resource reservation functions. In this paper, we present analytical, simulation and experimental results showing the impact of signaling system performance on quality of experience (QoE) for the potential users of multi-layer EuQoS system. In particular, the presented approach aims at ensuring user QoE of the connection setup phase by ensuring QoS for transferring signaling messages by the network.

Keywords—call setup delay, class of service, heterogeneous networks, next generation networks, quality of experience, quality of service, signaling system.

1. Introduction

This paper deals with an impact of the signaling system on user's quality of experience (QoE) in next generation networks (NGN). In particular we focus on signaling system and its procedures implemented within the EuQoS[1] project [1]. The aim of the EuQoS system [2], [3] is to guarantee end-to-end quality of service (QoS) in heterogeneous multi-domain networks. For this purpose the EuQoS system combines a complete architecture and a full framework [3] to provide absolute end-to-end QoS guarantees for a number of end-to-end classes of service (E2E CoS) [4], [5]. The architecture considers heterogeneous network scenario, addressing the multi-core IP network, as well as, the current access network technologies, such as WiFi (wireless fidelity), xDSL (digital subscriber line), LAN/Ethernet (local area network/Ethernet) and UMTS (universal mobile communication systems). Moreover, the generic architecture is open to add new network technologies. The applications, for which finally QoS is provided are, among others, voice over IP, video on demand, data transfer, and interactive game. All of them require call setup phase before data transfer.

The call setup phase is accomplished by the EuQoS signaling system, which has been designed in compliance with the ITU-T recommendations about signaling requirements for IP QoS networks [6] and requirements for resource and admission control functions, as defined in scope of NGN activities [7]. The signaling system follows a "push mode" approach of NGN architecture [8], i.e., the system requires that in order to start the setup procedure [9], the applications must send resource reservation requests in an explicit way to the control plane architecture.

The setup procedure starts at sending the calling user's request to the system and finishes when receiving the corresponding response indicating that the new call can be admitted and the called user is ready to initiate the connection. The time between these two events is well known as call setup delay [10]. In [11], ITU-T imposes limitations on expected values of setup delay. Among other requirements, it states that for international connections and under normal load conditions [12], the setup delay for the 95% of the setup procedures should be less or equal to 11 s.

In fact, the first experience of quality by the multimedia application users takes place during the request of a new call. If the request is not accepted or "slowly accepted", the confidence of the user decreases. The notion of QoE is already evolving. In the last years, the ITU-T Standardization Group 12 (SG12) specified the QoE requirements as definable end-to-end parameters, which provide information not only about the network layer (as performed by the QoS parameters) but also about the transport and application layers behavior [13].

We follow this notion of QoE and in our studies the definable end-to-end parameter is the call setup delay as suggested in [14]. The aim of this paper is to show the impact of signaling system performance on setup delay in EuQoS system based on NGN architecture.

In order to guarantee target values of call setup delay, the system should dedicate some resources to the signaling process (setup and release procedures). These resources are both server and network resources. The server resources are intended for the processing of the signaling messages within the signaling servers, which manage new calls. Whereas, the network resources are intended

[1]EuQoS – IST 6 FR EU project "End-to-end Quality of Service Support over Heterogeneous Networks".

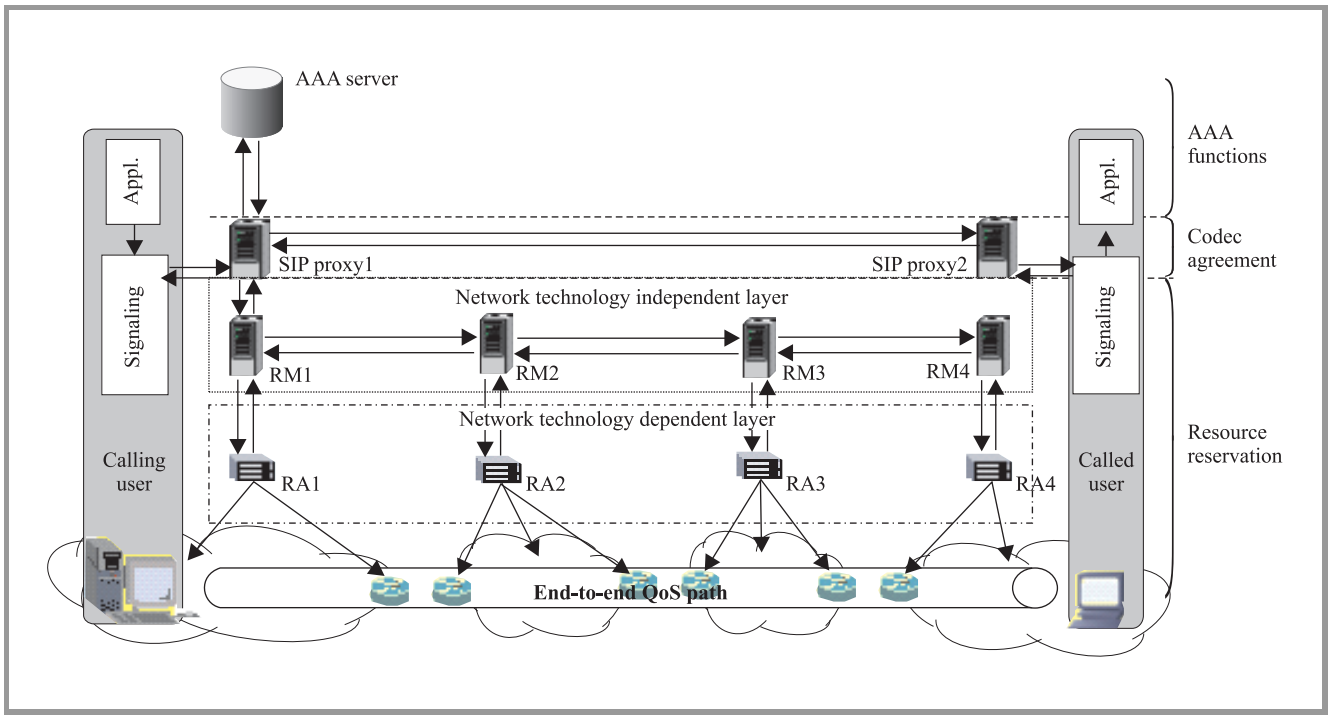


Fig. 1. The EuQoS system.

for transferring the signaling messages by the network between signaling servers. For reserving the necessary server and network resources, we should correctly dimension the signaling system. We propose to use the decomposition approach, which separately considers three phases of the signaling procedure in EuQoS system – all the operations performed:

- at the application layer, it means in session initiation protocol (SIP) proxies and authentication, authorization and accounting (AAA) servers for, e.g., voice over IP application;
- at the technology independent/technology dependent layer (TI/TD layer), for performing the admission control and resource allocation functions;
- at the network layer for transferring the signaling messages by dedicated signaling CoS.

Since the above three phases are distributed in time, i.e., they do not carry out simultaneously, we may enforce delay constrains for each one, so that the sum of the delay constrains of all the phases is less or equal to 11 s for the 95% of setup procedures. For illustration purposes, in our analysis we assume that the target setup delay value introduced by each of considered phases should be not greater than 3.5 s for the 95% of setup procedures.

Let us focus on the first phase of setup procedure, i.e., the operations performed at application layer. In previous studies of the EuQoS project [15], we obtained required performance parameters for the involved SIP proxy devices, which actually are less demanding than commercial ones. In fact, based on the literature we can observe that the SIP proxies are capable of running a high number of calls per second. For example, CISCO SIP proxies is able to handle about 100 call/s [16] and the direct routing approach allows even higher rates [17]. In the same way, the current AAA servers may manage until thousands of calls per second, e.g., the host intrusion detection system (HIDS) v3.1 of HP [18]. Measurements of mean establishment times of voice over IP calls related with AAA and SIP proxies functions were presented in [19]. Under normal load conditions, the mean establishment time is 0.94 s. Taking into account the state of the art, in our studies we focus on performance evaluation of the TI/TD layer and the transfer of signaling messages by the network (signaling class of service), which has been studied with less attention in the literature. These processes are the center of our studies in Sections 3 and 4, respectively. Previously, we present main features of the EuQoS system in Section 2. Finally, Section 5 summarizes the paper.

2. Overview of EuQoS System

The EuQoS system consists of three main layers, which are application layer, technology independent/technology dependent layer, and network layer as indicated in Fig. 1. The signaling procedure to set up a new connection (e.g., voice over IP) in the network is the following.

When any user sends a new call request to the EuQoS system, this initiates security functions with the AAA server.

These functions check if the user sending a call is authorized to use the EuQoS system. Next, the SIP proxy initiates the codec agreement with the called user. After the codec agreement, the first resource manager (RM) in the way checks whether the end-to-end QoS path to the called user exists and could provide the QoS guarantees required by the associated end-to-end CoS. The RM in the access domain periodically receives from the QoS routing protocol, the information about the QoS paths, i.e., the end-to-end paths with predefined QoS guarantees (in the form of target values for QoS parameters as IPTD, IPDV, IPLR). The QoS routing protocol is an enhanced version of border gateway protocol called EQ-BGP [2], [20], which builds end-to-end QoS paths on multi-domain network.

Afterwards, the system checks if, currently, there are available network resources (bandwidth, buffer capacities) to accept the call in consecutive domains. This is performed by the admission control functions implemented in resource managers and resource allocators (RA) as it will be indicated in the next section. If the process positively concludes, the system (SIP proxy) sends an affirmative response to the calling user allowing the communication.

Figure 1 presents the EuQoS signaling system, which is divided into the servers involved in AAA functions, the servers (SIP proxy) involved into the codec agreement and the servers (RM and RA) involved into the admission control function and resource allocation. The last ones, in turn, are divided into the servers, which are independent of the network technology (RM) and the servers, which functionalities depend on the network technology (RA). The purpose of this specialization is to allow each domain for custom implementation of its own QoS mechanisms without requiring other domains to be aware of their specific details.

Using the ITU-T terminology for NGN [21], [22], the SIP proxy would fulfill the service control function (SCF) at the service stratum; whereas, the resource manager and resource allocator would carry transport control function (TCF) at the transport stratum. The resource manager we would call policy decision functional entity (PD-FE), which is independent of the network technology and the resource allocator we would call the transport resource control functional entity (TRC-FE), which depends on the network technology.

A set of signaling elements (functions, databases, interfaces, etc.), as well as the signaling protocols involved in the different signaling layers were defined and implemented in the system. The diameter base protocol [23] communicates the SIP proxy with the AAA server in the access domain; the SIP protocol communicates end users and proxies, next steps in signaling (NSIS) protocol [24] connects the RMs and, at last, common open policy service (COPS [25]) communicates the RMs with the RAs and these ones with the network devices.

3. Performance Evaluation of TI/TD Layer

In this part, we analyze the impact of call handling scenario and the processing of messages at TI/TD layer on the part of the setup delay related to the TI/TD layer, which we denote as T_{ti-td}.

We focus on results of T_{ti-td} only for simulation studies, because the prototype implementation of RM and RA servers in the project was not optimized to take adequate conclusions. We present details about the call handling scenario at the TI/TD layer, a methodology to calculate T_{ti-td} for single and multi-domain scenario as well as the assumed simulation model. The aim of our studies is to show how many calls the system can handle respecting target values of T_{ti-td} (3.5 s for the 95% of setup procedures). In [15] we presented simulation results of performance evaluation on TI/TD layer. In this paper we enhance these studies showing detailed simulation results of analyzed queuing network.

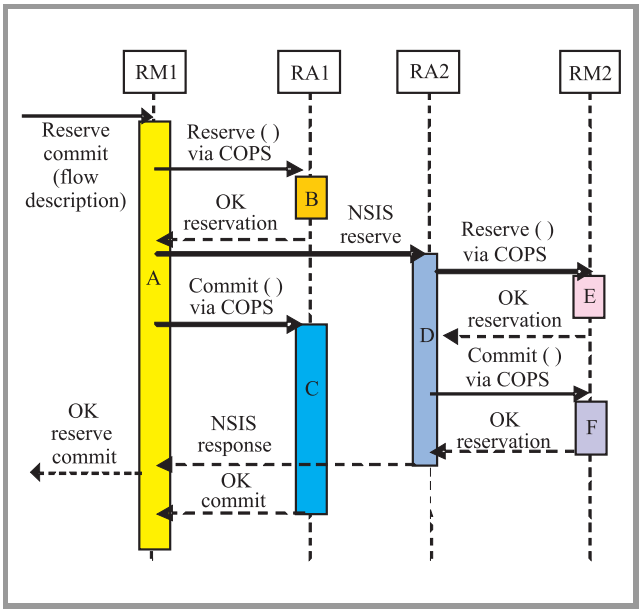


Fig. 2. Call scenario for two domains (domain 1: RA1, RM1; domain 2: RA2, RM2), one direction.

Figure 2 shows the horizontal (between RMs) and vertical (between RM and RA) signaling exchange for the successful setup of a call traversing two domains (domain 1 and domain 2). The call handling scenario is as follows: RM1 receives a QoS request and asks to the connection admission control (CAC) module in RA1 if there are enough resources to handle the new connection. If so, the requested resources are reserved and RA1 sends a confirmation to RM1. Next, RM1 forwards the QoS request to its peer RM2 and in parallel sends a request to RA1 to actually allocate the reserved resources in the associated access network equipment. When RM1 receives the confirmation from both RA1 and RM2, it replies to SIP proxy that the new call

can be admitted. As we can see in Fig. 2 the call handling scenario in RM2 and RA2 (i.e., in the egress domain) is the same as in RM1 and RA1 (ingress domain).

When the path has more than two domains, the resources are reserved also in transit domains. However, only the ingress domain RM checks whether an end-to-end path with feasible QoS exists. For bi-directional calls, the call handling process is performed in parallel during the setup procedure. So, in order to calculate the T_{ti-td} for a successful scenario, it is sufficient to simulate it only in one direction.

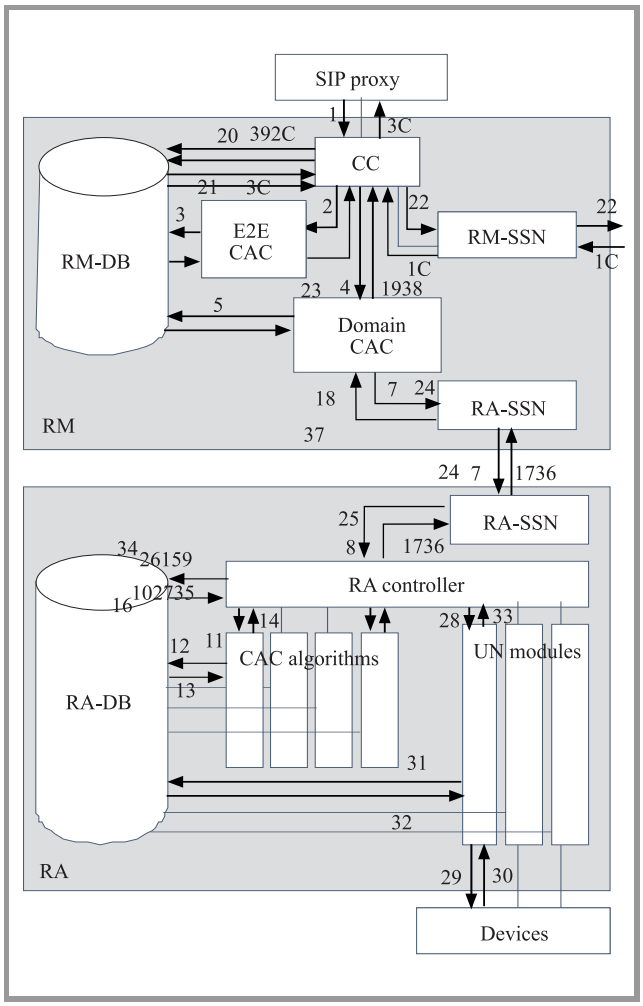


Fig. 3. The RM and RA architecture and call handling scenario for ingress domain calls and confirmations. Explanations: DB – data base, UN – underlining network, $1,2\ldots39$ task numbers, C – confirmation task.

The detailed architecture of the RM and RA, together with the tasks identifications (task ID) are presented in Fig. 3. In the RM, the call controller (CC) receives QoS requests and controls all the CAC submodules. The end-to-end CAC (E2E CAC) is in charge of checking whether an end-to-end path with feasible QoS characteristics actually exists (by looking at its EQ-BGP routing information base). Then, the CC asks the domain CAC to check the operator policies and to look for an intra-domain path. The domain CAC has two submodules, one for the intra-domain part and the other one for the inter-domain link. Finally, the RA controller receives the QoS request via signaling module interconnecting the RM with the RA signaling and service negotiation (RA SSN). The RA controller runs a different CAC algorithm for each CoS. The RA CAC algorithms check the amount of available resources by querying the RA data base (RA-DB) and decide about the acceptation of the new call. If the call is accepted, the RA controller asks the appropriate access network UN module to configure its network devices for handling the new call. Different mechanisms must be configured depending on the type of network device (IP router, LAN/Ethernet switch, WiFi access point, etc.). Note that UN modules are involved in the call handling process only in *access* domains. For *transit* domains, resources are reserved basing only on the decision of the RA CAC algorithm, and there is no dynamic configuration of resources in inter-domain network devices.

We model the RM and RA architecture (Fig. 3) as the queuing network shown in Fig. 4. The simulation model is composed of a chain of servers, each one associated to an infinite-length FIFO (first in first out) queue. We distinguish eight types of servers: RM-CC (1), RM-DB (2), RM-RA link (3), RM-RM link (4), RA-C (5), RA-DB (6), RM-RA link (7), devices (8).

We assume that new calls and call acceptance confirmations independently arrive to the RM according to Poissonian processes with given mean arrival rates. Service processing times are the following deterministic values:

- RM-CC, RA-CC: $t_{RM-CC} = t_{RA-CC} = 100\ \mu$s;
- database access in the RM and RA: $t_{RM-DB} = t_{RA-DB} = 1$ ms;
- RM-RA link, RA-RM link, RM-RM link: $t_{RM-RA} = t_{RA-RM} = t_{RM-RM} = 1$ ms;
- access to the device: $t_{DEV} = 600$ ms (WiFi) and $t_{DEV} = 100$ ms (IP).

The processing times for WiFi and IP devices are based on test bed network equipment (LINKSYS WiFi access point WRT54G, CISCO 1841 router or Linux router kernel 2.6.18 IMQ). For other elements as call controllers or data bases, we tried to infer the expected processing time based on today's high speed servers.

We consider three event types in the simulation model: *new call arrival, new confirmation arrival*, and *departure of a task from a server*. Arrivals and confirmations invoke a sequence of tasks, whereas, the departure of a task from one server determines its arrival to another server. Furthermore, we distinguish call events processed at *ingress, transit* or *egress* domains. The maximum number of tasks is performed when processing calls at the ingress domain, since the E2E CAC is checked. Fewer tasks are required to

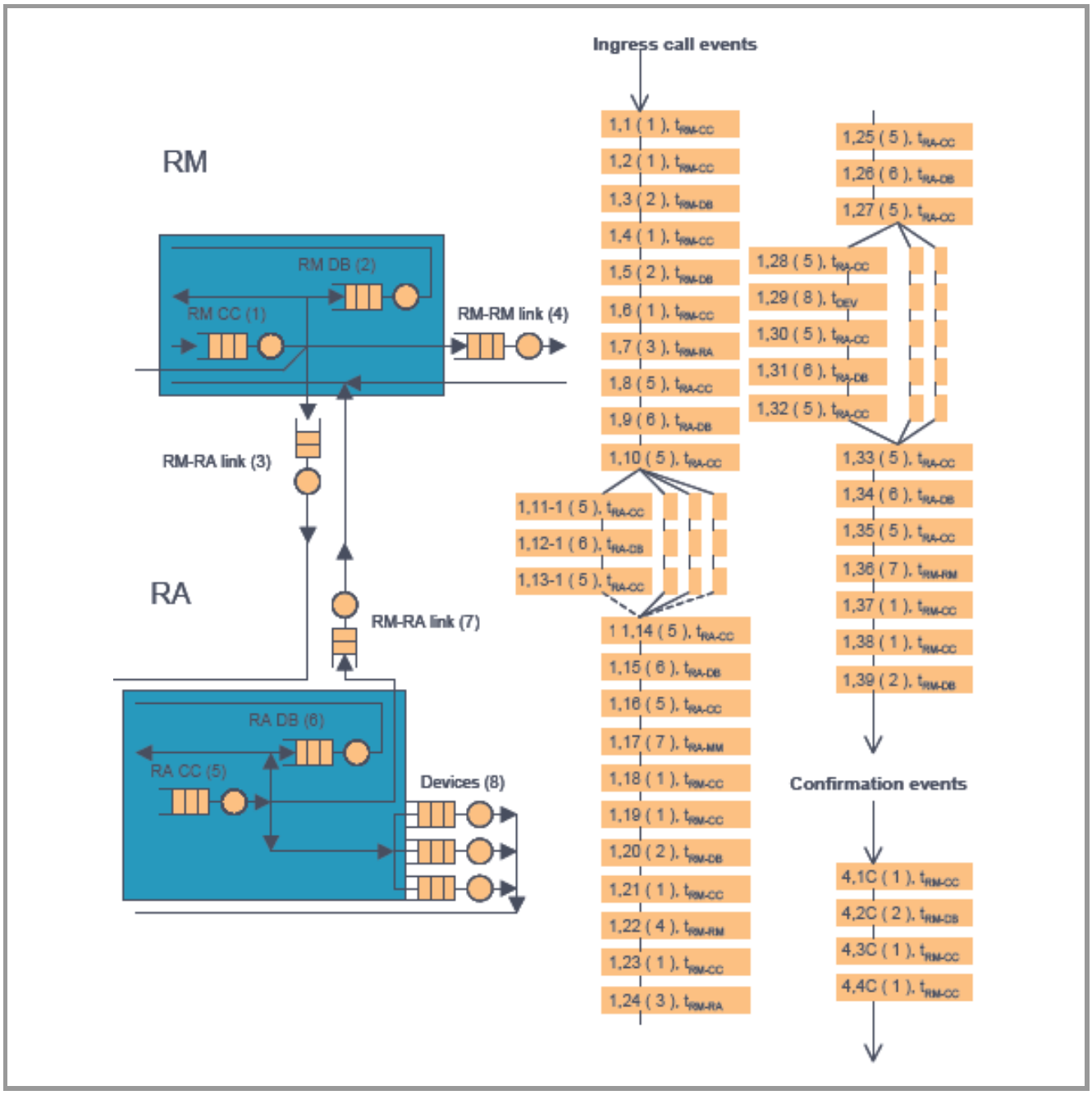


Fig. 4. The RM and RA for the ingress domain – simulation model.

process confirmation arrivals. The mean call arrival rates are $\lambda_{ingress}$, λ_{egress}, $\lambda_{transit}$, and the mean arrival rate of confirmations for ingress and transit domains are $\lambda_{conf-ingress}$ and $\lambda_{conf-transit}$. Due to space limitations, we only present events related to ingress domain.

Note that, in these simulations, we do not model the state machine of NSIS or COPS (see Fig. 2). The detailed message exchange of NSIS is modeled in simulation studies of signaling class of service (see Section 4). For COPS protocol, we model its performances by t_{RM-RA}, t_{RA-RM} times only.

The above simulation model has been coded in a discrete event simulator written in C++. Each event is represented by the quadruple <*Sequence ID, Task ID, Server ID, Processing time*>. The Sequence ID denotes whether the event is a call or a confirmation arrival.

We begin our studies with single domain performance evaluation. In particular, we simulate two scenarios. First, we assume that the single access domain handles both ingress and egress calls and confirmations assuming $\lambda_{ingress} = \lambda_{egress} = \lambda_{conf}$. We simulate WiFi and IP access domains in isolation. For each test we vary the call and confirma-

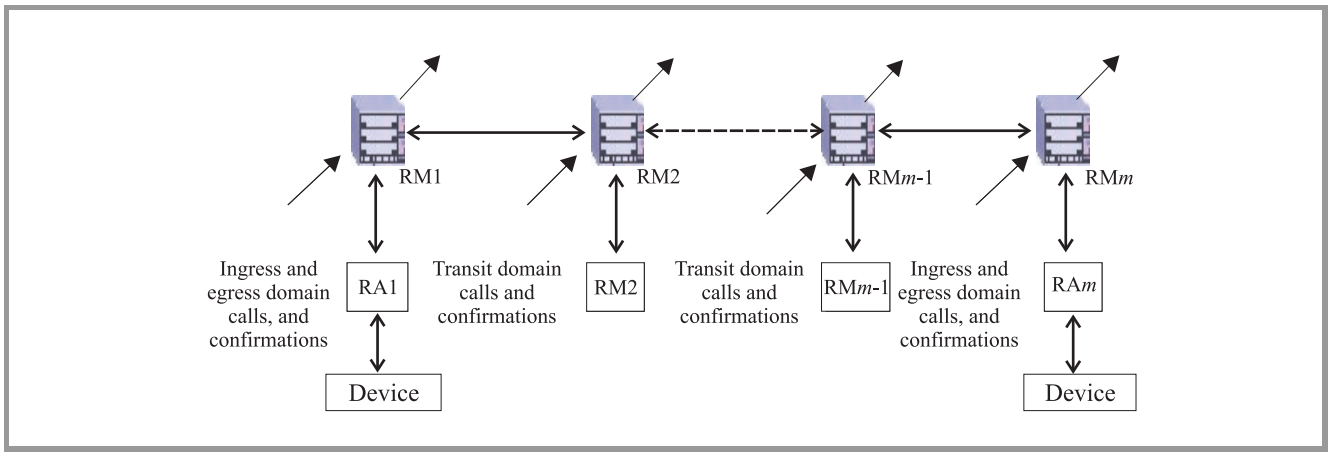


Fig. 5. Multi-domain call handling scenario with m domains, where $m = 2+n$, 2 access domains and n transit domains.

tion arrival rates up to the 95% resource utilization of the RM-RA elements (in this case the bottleneck element is the device itself).

In the next step, we analyze performances of single transit domain assuming that RM and RA handle only transit calls and confirmations assuming $\lambda_{transit} = \lambda_{conf}$; in this case the bottlenecks are RM-DB and RA-DB, it means the access to the databases.

Finally, based on the results obtained for single ingress/egress and transit domains, we are able to calculate the quantiles of T_{ti-td} for a multi-domain scenario (see Fig. 5).

For this purpose we distinguish among the following sequences of tasks performed by RM and RA in the ingress domain:

- $S1_{ingress}$: resource reservation tasks in RM1<−>RA1 performed before forwarding the request to the next RM;
- $S2_{ingress}$: tasks in RM1<−>RA1<−>DEV1, i.e., resource allocation tasks;
- $S3_{ingress}$: RM tasks invoked by confirmation arrival from neighboring domain.

Thus, we define the delay T_{ti-td} as follows:

$$T_{ti-td} = t_{ingress}^{S1} + \max\left\{t_{ingress}^{S2}, \left(n \cdot t_{transit} + t_{egress} + t_{ingress}^{S3}\right)\right\}, \quad (1)$$

where $t_{ingress}^{S\#}$ is the delay introduced by sequence S#, t_{egress} is the delay in the egress domain, n is the number of transit domains, $t_{transit}$ is the delay introduced by a transit domain.

Based on the detailed simulation results obtained for single domain scenarios (for ingress/egress and transit) we can derive:

$$t_{ingress}^{S2} < t_{ingress}^{S3} + t_{egress} + n \cdot t_{transit}. \quad (2)$$

By applying (1) and (2) the T_{ti-td} is calculated:

$$T_{ti-td} = t_{ingress}^{S1} + t_{ingress}^{S3} + t_{egress} + n \cdot t_{transit}. \quad (3)$$

In our studies, we calculate results when the access domain are both WiFi or both IP. The T_{ti-td} quantiles for 2, 10 and 20 domains is presented in Figs. 6–8.

In particular, Fig. 6 shows the results for two WiFi and two IP access domains. For both scenarios, the curves show a significant difference between the delays obtained for the two technologies. With reference to the 3.5 s target for the 0.95-quantile of T_{ti-td}, we observe that, in WiFi, $\lambda_{ingress}$, λ_{egress}, and λ_{conf} should not exceed about 0.6 call/s. In IP domain, the limit is about 4.75 call/s. For scenarios with 10 and 20 domains we assume that two domains are access domains while the others are transit domains.

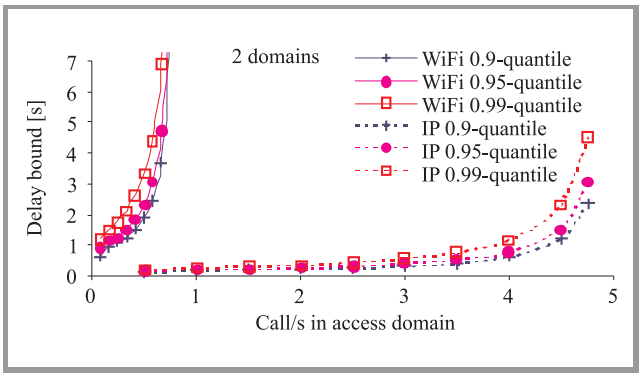


Fig. 6. Quantiles of T_{ti-td} for 2 WiFi or 2 IP domains versus $\lambda_{ingress}$ ($\lambda_{ingress} = \lambda_{egress} = \lambda_{conf}$).

Figures 7 and 8 show the quantile values of T_{ti-td} for WiFi and IP access domains, respectively. The transit call rates are 233 or 300 call/s in each transit domain. The results show that the number of transit domains has smaller impact on the T_{ti-td} than the transit call arrival rate. Moreover,

acceptable call arrival rates are larger for IP access domains than for WiFi.

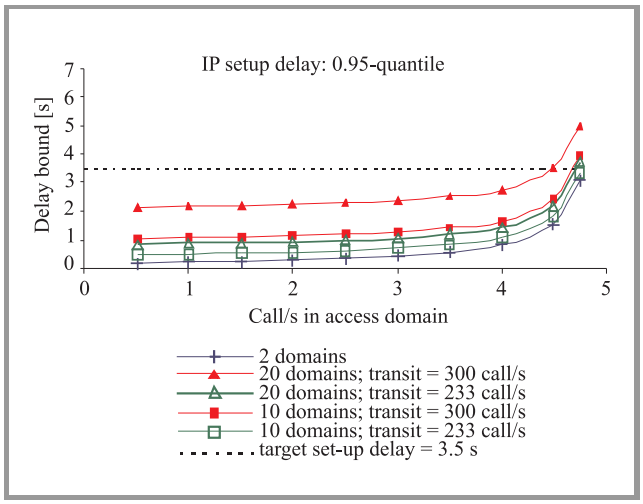


Fig. 7. The 0.95-quantile of T_{ti-td} for 2 IP access domains versus $\lambda_{ingress}$ ($\lambda_{ingress} = \lambda_{egress}$ and $\lambda_{ingress} = \lambda_{conf}$).

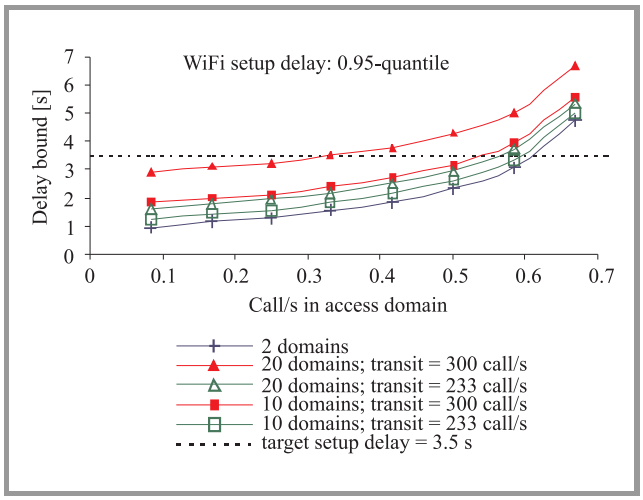


Fig. 8. The 0.95-quantile of T_{ti-td} for 2 WiFi access domains versus $\lambda_{ingress}$ ($\lambda_{ingress} = \lambda_{egress}$ and $\lambda_{ingress} = \lambda_{conf}$).

Figure 9 shows some characteristics of the transit RM and transit RA. In particular, we collected characteristics of server utilization, mean delay in the queues, and mean queue sizes. As we can observe, the main bottlenecks of RM and RA elements for transit domains are RM-DB and RA-DB servers. The characteristics of these two servers are very close to each other.

Concluding, the signaling system at the TI/TD level is able to handle about 300 call/s in transit domains and it correctly scales with the number of transit domains (for values of 10 and 20 transit domains). An important open point is the call arrival process for NGN multi-service networks.

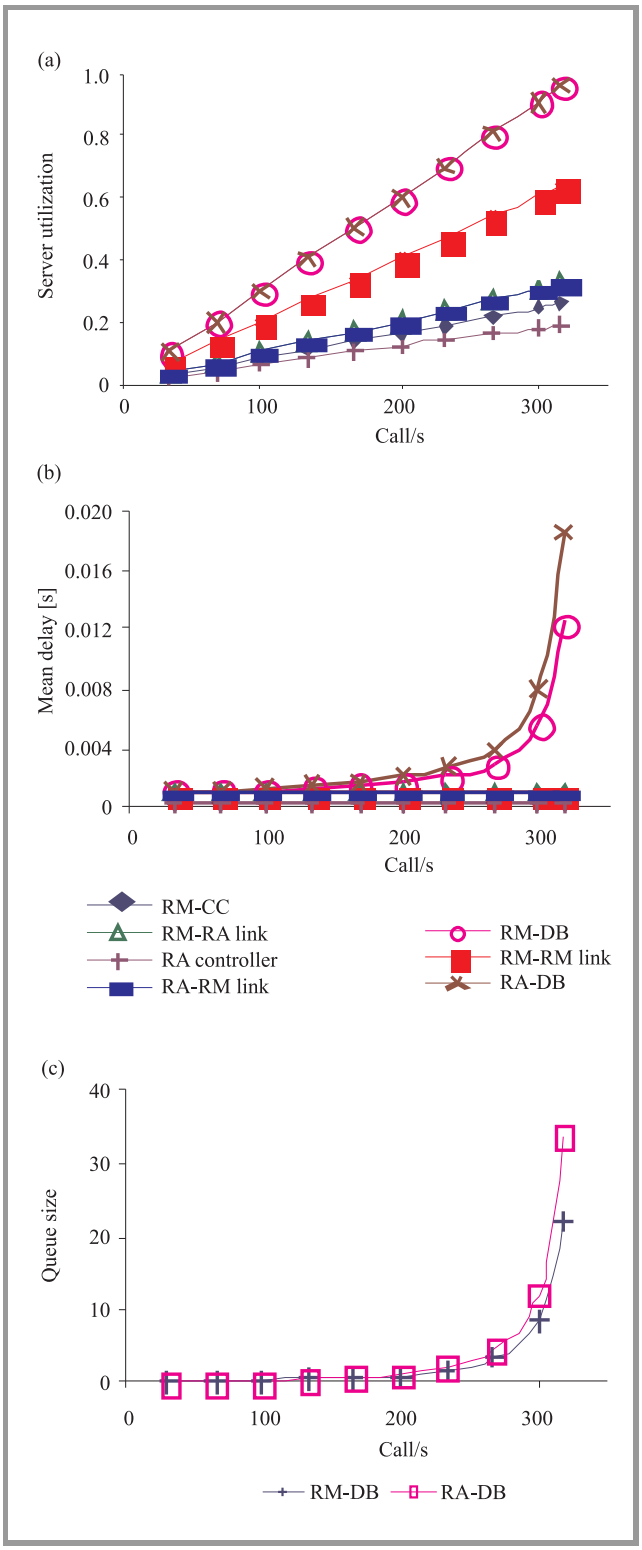


Fig. 9. Transit domain characteristics: (a) utilization of RM and RA servers, (b) mean delay, and (c) mean queue size for RM and RA elements versus transit call arrival rates ($\lambda_{transit}$) $\lambda_{transit} = \lambda_{conf-transit}$.

In these studies, we commonly considered telephony arrival model only but, in further studies, we are also facing up multi-service models.

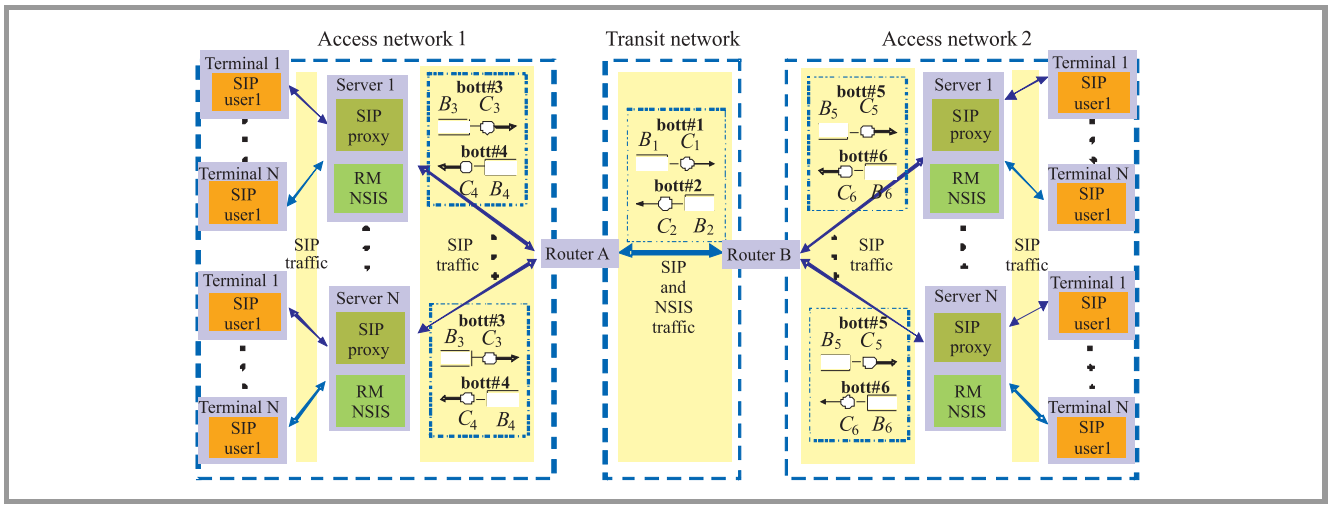


Fig. 10. Model of the EuQoS signaling system.

4. Performance Evaluation of the Signaling Class of Service

At the network layer we propose to implement the signaling class of service. The IETF document [4] defines, among others, the signaling class of service (S-CoS) designated to carry signaling traffic related with setup and release procedures. The objective of the S-CoS is to ensure target values of the part of the call setup delay related to the transfer of the signaling packets. We call this delay as *transfer packet call setup delay* or briefly T_s *delay*. We expect to obtain target values of T_s *delay* by transferring the signaling packets with adequate quality of service expressed in target maximum IPTD. In fact, according to [4] we can assume that the S-CoS tolerates the delay variation of the signaling traffic. One of the solutions to guarantee the QoS objectives for the signaling traffic is a correct resource provisioning [26].

For the aim of provisioning the S-CoS, we investigate the bottlenecks within the signaling path. The bottlenecks are the slowest links [27] in whose entrance packets of different setup procedures gather and where we should provision appropriate resources for each of the setup procedures.

To compute the necessary amount of bandwidth within the S-CoS for one setup procedure, we propose to ensure the same maximum IPTD (maxIPTD) in all the bottlenecks along the path. From the knowledge of the sizes of packets submitted to each bottleneck, and more precisely from the length of the longest burst of packets of the setup procedure submitted to each bottleneck we obtain the maxIPTD for 1 isolated setup procedure as (4)

$$\max IPTD = \frac{length\ of\ longest\ packet\ burst_{bott\,1} \times 8}{C_{bott\,1}} = \ldots = \frac{length\ of\ longest\ packet\ burst_{bott\,i} \times 8}{C_{bott\,i}}. \quad (4)$$

The value of $C_{bott\,i}$ indicates the necessary amount of bandwidth for one setup procedure in the bottleneck i of S-CoS.

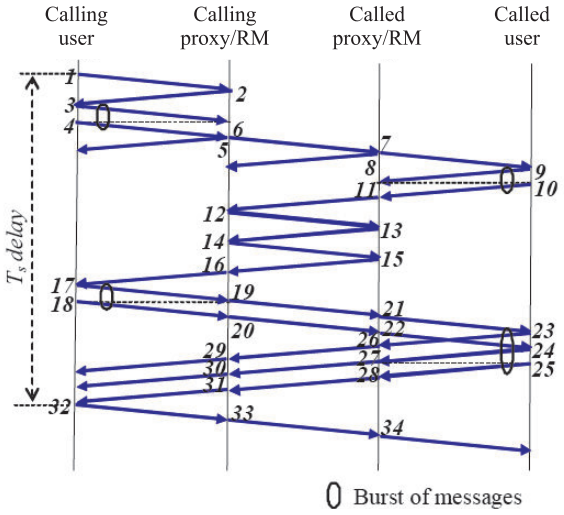
The equation, which completes the consistency of the system of linear equations presented in (4), comes from the value of T_s *delay*. In fact, we defined that the setup procedure should finish in a time equal to 3.5 s as presented in equation:

$$T_s\ delay = \sum_{all\ bottleneck\ s} \frac{\sum_{all\ packets\ in\ bottleneck} packet\ length \times 8}{C_{bott\,i}}. \quad (5)$$

In the EuQoS system we assume that bottlenecks only may appear at the exit of SIP proxies and next inter-domain border routers as presented in the model of Fig. 10. Therefore, we consider only SIP and NSIS traffic, which is the unique signaling traffic in these bottlenecks.

As we may deduce from Eqs. (4) and (5), the necessary resources for signaling traffic ($C_{bott\,i}$) depend on the message sequence. Figure 11 shows the setup message sequence in the EuQoS system for which we implement the S-CoS. The lengths of the packets referred in Fig. 11 do not consider user data protocol (UDP), IP and link layer headers. Based on the above setup message sequence and assumptions and by the way of example, we will present further down simulation and measurement results.

Since packet losses cause retransmissions and, as an undesired result, increase delay of the whole setup procedure, we provision the buffer resources to ensure no losses in the queues (IPLR = 0) over normal operation of the network (no link failures). Since the biggest burst of packets in one setup procedure (EuQoS system) contains 2 packets (see Fig. 11) in the direction from calling to called and 3 packets from called to calling, then, the buffer size for M simultaneous setup procedures should equal $2 \times M$ packets in the calling-to-called direction and $3 \times M$ packets in the opposite direction.



SIP messages:

Reference	Name	Message length
1	Invite	706 Bytes
2	Auth. Req.	501 Bytes
3,32,33,34	Ack	246 Bytes
4	Invite Auth.	902 Bytes
5,8,9	Trying	216 Bytes
6,7	Invite	902 Bytes
10,13,16	Progress	756 Bytes
17,19,21	Prack	315 Bytes
18,20,22	Update	737 Bytes
23,26,29	200 OK. Prack	260 Bytes
24,27,30	200 OK	262 Bytes
25,28,31	Ringing	444 Bytes

NSIS messages:

Numer ref.	Name	Message length
11,14	Reserve	702 Bytes
12,15	Response	212 Bytes

Fig. 11. SIP and NSIS message sequence in EuQoS system.

Let us remark that the transport protocol used to carry the signaling messages is not excessively important since, by provisioning the buffers, we assume no losses. Anyway, if we transport the signaling messages using the UDP protocol, we have to pay attention to the retransmission timers of the signaling protocols because these values are manually set and do not consider the round trip time (RTT) values of the network. The SIP protocol defines the default value of the retransmission timer equal to 500 ms when we use UDP protocol [28; Section 17.1.1.1]. The timers in the calling user will stop when the messages and its responses cross the whole network. This time is usually higher than 500 ms, so, such a value of retransmission timer results in the unnecessary retransmission of signaling packets (packets are not lost). This could cause a dangerous overload of the S-CoS. Therefore, in EuQoS implementation, we set the default values of SIP timers over UDP to 3 s.

4.1. Evaluation by Simulations

The performed tests aim at validating the proposed provisioning method to ensure target T_s *delay*. In our simulations we assume the same model as presented in Fig. 10. Moreover, we assume that the value N is equal to 5, this implies 5 servers and 25 terminals in each access network. The terminal i of the access network 1 (AN1) initiates a setup procedure with the terminal i of the access network 2 (AN2) and also the terminal i of the AN2 initiates another setup procedure with the terminal i of the AN1. All the terminals (50) initiate setup procedures at the same time and when the setup procedure corresponding to the terminal i finishes, then, the terminal i instantly initiates a new setup procedure. This is the worst-case scenario because finally, any new setup procedure finds the system full. Note that, in a real scenario, the setup procedures randomly arrive to the system and may find the system not completely full. Therefore, the values of T_s *delay* presented in these simulations are the upper bound.

Table 1
Bottleneck link capacity for 1 unique setup procedure

Initiated in AN1						
C [kbit/s]	C_1	C_2	C_3	C_4	C_5	C_6
bott#1–bott#6	38.3	32.3	49.6	32.3	38.3	42.3
bott#1–bott#4	28.7	24.2	37.2	24.2	–	–
bott#1–bott#2	15.8	13.3	–	–	–	–
Initiated in AN2						
bott#1–bott#6	32.3	38.3	42.3	38.3	32.3	49.6
bott#1–bott#4	21.4	25.3	28.0	25.3	–	–
bott#1–bott#2	13.3	15.8	–	–	–	–

Table 2
Bottleneck buffer size for 1 unique setup procedure

Initiated in AN1						
B [packets]	B_1	B_2	B_3	B_4	B_5	B_6
bott#1–bott#6	2	3	2	3	2	3
bott#1–bott#4	2	3	2	3	–	–
bott#1–bott#2	2	3	–	–	–	–
Initiated in AN2						
bott#1–bott#6	3	2	3	2	3	2
bott#1–bott#4	3	2	3	2	–	–
bott#1–bott#2	3	2	–	–	–	–

Table 3
Link capacities for 25 setup procedures initiated in AN1 and 25 initiated in AN2 (values set in the simulation scenario)

C [kbit/s] / B [packets]	C_1 / B_1	C_2 / B_2	C_3 / B_3	C_4 / B_4	C_5 / B_5	C_6 / B_6
bott#1–bott#6	$25\times(38.3+32.3)=$ **1765.0** / $25\times(2+3)=$ **125**	$25\times(32.3+38.3)=$ **1765.0** / $25\times(2+3)=$ **125**	$5\times(49.6+42.3)=$ **459.5** / $25\times(2+3)=$ **125**	$5\times(32.3+38.3)=$ **353.0** / $25\times(2+3)=$ **125**	$5\times(38.3+32.3)=$ **353.0** / $25\times(2+3)=$ **125**	$5\times(42.3+49.6)=$ **459.5** / $25\times(2+3)=$ **125**
bott#1–bott#4	$25\times(28.7+21.4)=$ **1252.5** / $25\times(2+3)=$ **125**	$25\times(24.2+25.3)=$ **1237.5** / $25\times(2+3)=$ **125**	$5\times(37.2+28.0)=$ **326.0** / $25\times(2+3)=$ **125**	$5\times(24.2+25.3)=$ **247.5** / $25\times(2+3)=$ **125**	**100 000** / **125**	**100 000** / **125**
bott#1–bott#2	$25\times(15.8+13.3)=$ **727.5** / $25\times(2+3)=$ **125**	$25\times(13.3+15.8)=$ **727.5** / $25\times(2+3)=$ **125**	**100 000** / **125**	**100 000** / **125**	**100 000** / **125**	**100 000** / **125**

The simulation environment is ns-2 [29] where we developed modules to simulate the signaling process and integrated them into the EuQoS simulation model [30]. The integration of these modules permits its use in any network scenario (even in environments with different network techniques). In our own-implemented modules we model the signaling protocols: SIP and NSIS.

We introduce a bottleneck in the respective link by setting the appropriate value of the link capacity obtained from the provisioning method presented above. The links that are not considered bottlenecks in a certain test are set to 100 Mbit/s. Also the links between terminals and servers are 100 Mbit/s links. Next, we perform three tests, each one with 10 000 setup procedures. The number of bottlenecks change from one test to another. The first test considers 6 bottlenecks, from bott#1 and bott#6 as indicated in Fig. 10. The next test considers 4 bottlenecks (from bott#1 to bott#4) and the last one considers only the bottlenecks in the both directions of the link between router A and router B, i.e., bott#1 and bott#2.

We expect that each bottleneck will require a different provisioning, since the volume of signaling traffic submitted to them is different (see Fig. 11). Tables 1 and 2 show the necessary values of capacity in the bottleneck links and size of the bottleneck buffers for one unique setup procedure initiated in the AN1 and one unique setup procedure initiated in AN2 for the cases of 6, 4 and 2 bottlenecks in the signaling path. Values of this tables we exploit to complete Table 3, which presents the values set in the links and buffers of the simulation scenario considering that all the 50 terminals initiate setup procedures.

For each test, we calculate the time that packets last on transferring the 100 Mbit/s links. This time is not considered within the provisioning method and we subtract it from the total T_s *delay* value obtained in each test. The T_s *delay* values of the test (after subtraction) are presented in Table 4. Specifically, Table 4 shows the T_s *delay* of the shortest (min) and longest (max) setup procedure. In our simulations, we discard the first setup procedures of the tests because, at the beginning, the system is empty and these first setup procedures finish earlier, i.e., we only consider the T_s *delay* values of the setup procedures after striking the balance of the simulation process. We may observe that the values of T_s *delay* respect the target value equal to 3.5 s in all the cases. In opposition to the simulations performed in [26], in the presented simulation approach there is no multiplexing gain in the network because the system is permanently occupied. In fact, the setup procedures arriving to the system synchronize ourselves and transfer the network in the same order. This simulation approach allowed us to validate the provisioning method presented above.

Table 4
Value of T_S *delay* for the scenarios with 6, 4 and 2 bottlenecks (simulation results)

Bottleneck	bott#1–bott#6 min/max	bott#1–bott#4 min/max	bott#1–bott#2 min/max
T_s *delay* [s]	3.499/3.500	3.499/3.500	3.500/3.500

4.2. Evaluation by Measurements

In this section we present measurement results of S-CoS performance evaluation. The aim of the measurements is to demonstrate that the S-CoS provisioned by the method presented in this paper may ensure target T_s *delay* in the test bed environment. On the other hand, we will also demonstrate that we cannot ensure delay requirements for the signaling traffic not carried by the S-CoS, if the network is heavy loaded.

Figure 12 presents the measurement test bed, where we emulate a scenario with two domains that are interconnected by an inter-domain link. The capacity of the inter-domain link equals 8 Mbit/s, while links inside each domain offer 100 Mbit/s capacity.

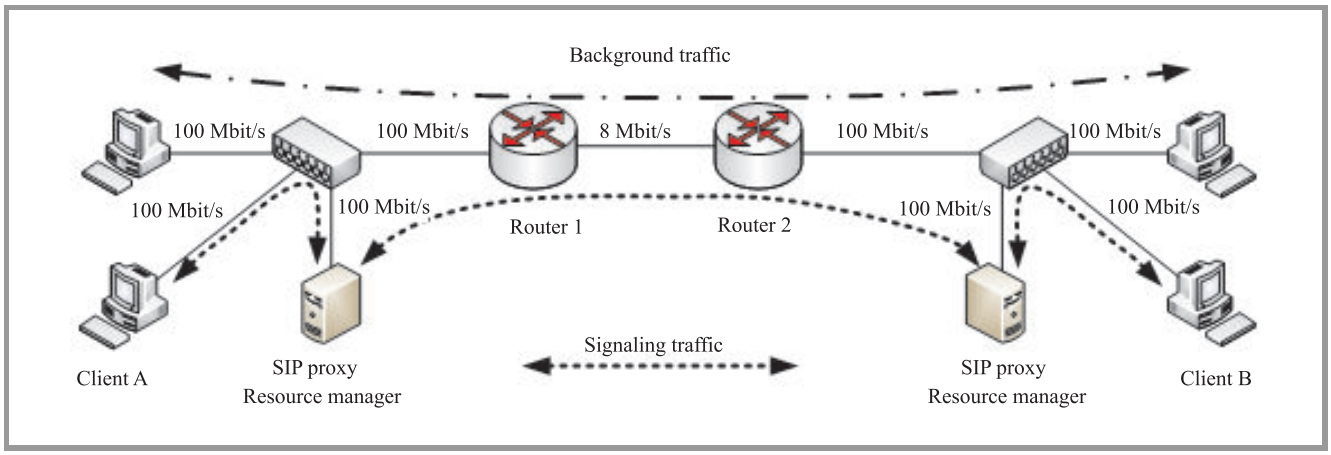


Fig. 12. Measurement test bed.

For implementing the S-CoS, all the signaling servers must "mark" their own packets before sending them into the network. This is done by setting the differentiated services code point (DSCP) field in the IP header of packets. The DSCP value assumed for the signaling packets is equal to binary value 101000 [4].

Figure 13 presents the model of the outgoing port of two routers: router 1 and router 2. In router 1 it is implemented in the direction from client A to client B and, in router 2, in the opposite direction.

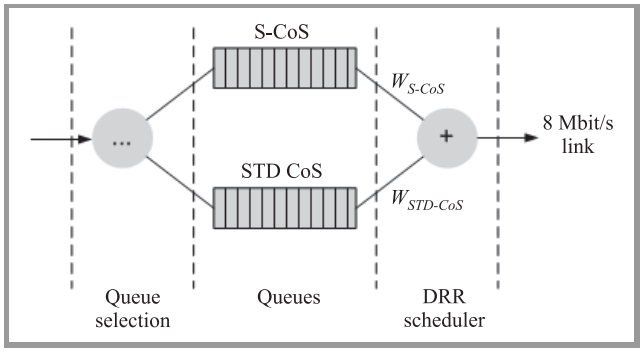


Fig. 13. Model of the outgoing port of the routers.

For our measurements, we consider only two classes of service: the S-CoS and the standard CoS (STD CoS), which is the best effort service. To ensure bandwidth separation between considered classes in IP router we use the deficit round robin (DRR) scheduler. We provision the S-CoS with 1000 and 2000 kbit/s in the two tests, respectively. The scheduler's weight for S-CoS traffic (w_{S-CoS}) is set to guarantee assumed amount of bandwidth, whereas, the scheduler's weight for STD CoS traffic ($w_{STD-CoS}$) complements the link capacity up to 8 Mbit/s.

By the provisioning method, we may calculate how many simultaneous setup procedures can be admitted, as a maximum, in the S-CoS not to exceed 1000 and 2000 kbit/s. When applying the provisioning method, we consider one unique bottleneck between proxy/RMs in the direction from client A to client B since the setup procedures are initiated only in client A. Moreover, we do not consider bottlenecks between clients and proxy/RMs because there is only signaling traffic. Table 5 presents the maximum number of simultaneous setup procedures for the provisioned S-CoS capacities, as well as the necessary buffer size set in the S-CoS to avoid losses for this number of simultaneous setup procedures.

Table 5
Class of service capacities and buffer sizes
(experiment configuration)

S-CoS capacity	1000 kbit/s	2000 kbit/s
Maximum number of simultaneous setup procedures	131	262
S-CoS buffer size	262 packets	524 packets

An artificial call generator in the client A initiates the setup procedures and maintains a constant number of them running in the system. When a setup procedure finishes, the call generator instantly initiates a new one.

On the other hand, the background traffic (STD CoS) is composed by 10 TCP connections in each direction, which fill the 8 Mbit/s link. We set the buffer size in the STD CoS equal to 1000 packets, which is sufficiently large to prevent any packet loss and introduces a high delay in the STD CoS traffic.

All measurements that we take, are performed at the signaling application level (T_s *delay*). We simplified the RM implementation with main focus on processing of signaling messages in order to reduce the delay introduced by state management operations, e.g., database access and storage and policy algorithms.

In the first experiment we compare the performance of signaling system with and without the S-CoS. We measure the T_s *delay* for cases with 131 or 262 running setup procedures, with capacity assigned to S-CoS equal to 1 Mbit/s or 2 Mbit/s, respectively. In Fig. 14 we show the maximum and minimum values of T_s *delay* measured in the test bed. The measurement consisted of at least 1000 setup procedures after warm up period.

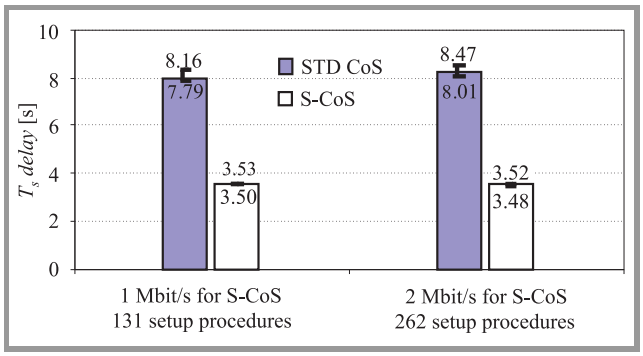


Fig. 14. Measurements of T_s *delay* within the EuQoS test bed.

By introducing the S-CoS in the network we are able to guarantee the value of the T_s *delay*; for both considered cases it is lightly higher than assumed target of 3.5 s. The tiny differences are due to the transfer of signaling packets by the not-bottleneck links. On the other hand, when signaling messages shares the capacity with the background traffic in STD CoS, the observed average T_s *delay* exceeds 8 s. As we expected, without the implementation of the S-CoS, we are not able to ensure the setup delay requirements for Internet telephony presented in [11]. Therefore, we argue that signaling traffic and best effort traffic must not be handled by the same network resources when the network is heavily loaded.

In the next experiment we evaluate the characteristics of T_s *delay* when the number of simultaneous setup procedures is lower than the calculated maximum value (for given S-CoS capacity). Figure 15 shows this characteristic for the case when 2 Mbit/s capacity is dedicated for S-CoS; we may see maximum and minimum measured values of T_s *delay*. The relation between T_s *delay* and the number of simultaneous setup procedures is linear, as we could expect. Moreover, for given number of simultaneous setup procedures, the T_s *delay* of different setup procedures does not vary so much (intervals are very small).

On the basis of the above results, we strongly recommend to introduce the dedicated class of service to ensure QoS

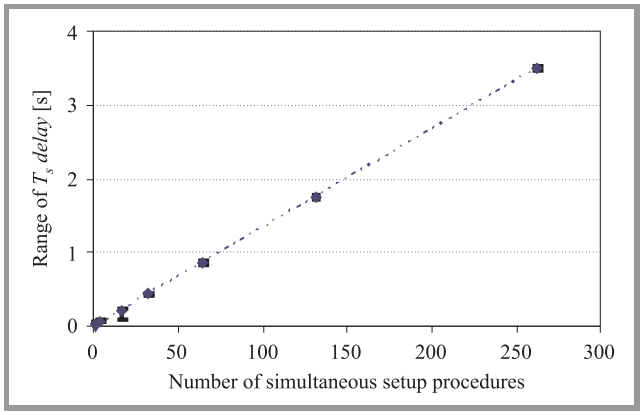


Fig. 15. Measured characteristics of T_s *delay* within the EuQoS test bed.

at the packet transfer of signaling messages. Unfortunately, the inherent characteristics of the signaling traffic (bursts of packets, sporadic traffic, etc.) do not allow to effectively control it and, because of this, it is not possible to carry the signaling traffic by other CoSs on aggregation with other kinds of traffics.

5. Conclusions

In this paper we studied the signaling performance of one example of next generation network system: the EuQoS system. We considered main processes inside the signaling system, as call handling scenario and transferring of the signaling messages by the network. Both these factors may affect into the call setup delay and, as a consequence, into the user quality of experience. We decided to confront them separately, using decomposition approach. As a result, we achieved a system capable of establishing end-to-end connections with the same requirements for call setup delay as assured in the ISDN network [10].

The key for the correct design of the system lies in reserving enough resources, both for signaling servers (processing power) and for the transfer of signaling messages in the network (bandwidth and buffer). For the signaling servers, we calculated the maximum call arrival rate (number of setup procedures per second) that the system can handle to assure given call processing delay. We considered all the signaling servers involved into the signaling system, paying special attention to the signaling servers designed to the resource reservation and allocation process over exemplary technologies. Following the analysis and relations between signaling servers, we evaluated system performance by means of dedicated simulation and measurement tools.

To complete the provisioning of resources, we investigated how the traffic generated by the signaling servers is handled in the network. We assumed that the number of simultaneous setup procedures handled by the signaling system is limited. Thanks to this assumption, we were able to provide dimensioning rules for signaling class of service. The rules provide a value of bandwidth and buffer space that should be dedicated for signaling traffic at the bottleneck points in order to guarantee given setup delay requirements. The devised provisioning method was verified by the simulation of the signaling system and by measurements in the test bed scenario.

Though we strived to do general analysis of the signaling system features, some results may be particular for the EuQoS system. Anyway, the presented proposals and methods can be used as a good guideline for the design of other signaling systems.

Future works in this field should be directed to find solutions for signaling congestion control, i.e., how to ensure a maximum number of simultaneous setup procedures within the system. It is also important to consider traffic models at call level, however, it requires an insight into measurements in multiservice networks. Other problem is related with the inherent complexity of the signaling systems in next generation networks. In this work, we con-

sidered only the main signaling servers in each domain, but in reality we should expect multiple ways of system decomposition and distribution.

Acknowledgements

We would like to thank all the partners of the EuQoS consortium for their cooperation and their work on developing the EuQoS system.

References

[1] "The EuQoS Consortium" [Online]. Available: http://www.euqos.eu/

[2] X. Masip-Bruin *et al.*, "The EuQoS system: a solution for QoS routing in heterogeneous networks", *IEEE Commun. Mag.*, vol. 45, no. 2, pp. 96–103, 2007.

[3] W. Burakowski *et al.*, "Provision of end-to-end QoS in heterogeneous multi-domain networks", *Ann. Telecommun. Springer*, vol. 63, iss. 11, p. 559, 2008.

[4] J. Babiarz and F. Baker, "Configuration Guidelines for DiffServ Service Classes". Internet RFC 4594, Aug. 2006.

[5] K. Chan, J. Babiarz, and F. Baker, "Aggregation of Diffserv Service Classes". Internet RFC 5127, Febr. 2008.

[6] "Signaling Requirements for IP QoS", ITU-T TR Q-Series Supplement 51 (10/2004).

[7] "Resource and admission control functions in next generation networks", ITU-T Rec. Y.2111 (09/2006).

[8] "General overview of NGN", ITU-T Rec. Y.2001 (10/2004).

[9] "Framework for the network management of IP-based networks", ITU-T Rec. E.417 (02/2001).

[10] "Network grade of service parameters and target values for circuit-switched services in the evolving ISDN", ITU-T Rec. E.721 (05/1999).

[11] "Network post-selection delay in PSTN/ISDN networks using Internet telephony for a portion of the connection", ITU-T Rec. E.671 (03/2000).

[12] "Network grade of service parameters and target values for maritime and aeronautical mobile services", ITU-T Rec. E.774 (10/1996).

[13] "Quality of experience requirements for IPTV services", ITU-T Rec. G.1080 (06/2008).

[14] A. Moorsel, "Metrics for the Internet age: quality of experience and quality of business", in *Fifth Perform. Worksh.*, Nuremberg, Germany, 2001.

[15] H. Tarasiuk *et al.*, "Designing the simulative evaluation of an architecture for supporting QoS on a large scale", in *Proc. QoSim 2008 Conf.*, Marseille, France, 2008 (to appear ACM Digital Library).

[16] CISCO systems document [Online]. Available: http://www.cisco.com/en/US/tech/tk652/tk701/-technologies_white_paper09186a00800a9818.shtml.

[17] V. Matic, I. Franicevic, and D. Sekalec, "Parallel SIP proxy servers using direct routing approach", in *Proc. Int. Conf. Softw. Telecommun. Comput. Netw. SoftCOM 2006*, Split-Dubrovnik, Croatia, 2006.

[18] "Host Intrusion Detection System (HIDS) v3.1. Sizing Guidelines and Tuning Primer", HP Company internal document, Sept. 2005.

[19] Y. Bai *et al.*, "A novel distributed wireless VoIP server based on SIP", in *Proc. Multimed. Ubiquit. Eng. Conf. MUE'07*, Seoul, Korea, 2007.

[20] A. Beben, "EQ-BGP: an efficient inter-domain QoS routing protocol", in *Proc. 20th IEEE Int. Conf. Adv. Inform. Netw. Appl.*, Los Alamitos, USA, 2006, pp. 560–564.

[21] N. Morita, H. Imanaka, O. Kamatani, T. Oba, and K. Tanida, "Overview and status of NGN standardization activities at ITU-T", *NTT Tech. Rev.*, Nov. 2007.

[22] J. Song *et al.*, "Overview of ITU-T NGN QoS control", *IEEE Commun. Mag.*, vol. 45, no. 9, pp. 116–123, 2007.

[23] P. Calhoun *et al.*, "Diameter Base Protocol". Internet RFC 5127, Sept. 2003.

[24] R. Hancock *et al.*, "Next Steps in Signaling (NSIS): Framework". Internet RFC 4080, June 2005.

[25] D. Durham *et al.*, "The COPS (Common Open Policy Service) Protocol". Internet RFC 2748, Jan. 2000.

[26] J. Mongay Batalla and R. Janowski, "Provisioning dedicated class of service for reliable transfer of signaling traffic", in *Proceedings 20th International Teletraffic Congress, June 2007, Ottawa, Canada*, Lecture Notes in Computer Science, vol. 4516. Berlin-Heidelberg: Springer: 2007, pp. 853–864.

[27] P. Rodriguez and E. W. Biersack, "Dynamic parallel access to replicated content in the Internet", *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 455–465, 2002.

[28] J. Rosenberget *et al.*, "SIP: Session Initiation Protocol". Internet RFC 3261, June 2002.

[29] "The Network Simulator ns-2" [Online]. Available: http://www.isi.edu/nsnam/ns/

[30] "The ns-2 signaling modules" [Online]. Available: http://tnt.tele.pw.edu.pl/include/tools/sim-euqos-ptl.tgz

Jordi Mongay Batalla was born in Barcelona, Spain, in 1975. He received the M.Sc. degree in telecommunications from Universitat Politècnica de València in 2000. He worked one year in Centro Nazionale di Astrofisica in Bologna, Italy, as research scientist specializing in qualities and capacities of diffserv networks. Nowadays, he is finishing Ph.D. studies in the Warsaw University of Technology, Poland. He is with Telecommunications Network Technologies (TNT) Group from 2004. His research interest focus mainly on quality of service in diffserv networks and next generation network architecture.
e-mail: jordim@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Jarosław Śliwiński received the M.Sc. and Ph.D. degrees from the Warsaw University of Technology, Poland, in 2003 and 2008, respectively. His research interests cover traffic control, systems' design and implementation methodology.

e-mail: jareks@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Halina Tarasiuk received the M.Sc. degree in computer science Szczecin University of Technology, Poland, in 1996 and Ph.D. degree in telecommunications from the Warsaw University of Technology, in 2004. From 1998 she is with Telecommunication Network Technologies Group at the Institute of Telecommunications, Warsaw University of Technology. In 2003 as a member of the group she received Rector's Award for scientific achievements. From 2004 she is an Assistant Professor at the Warsaw University of Technology. From 1999 to 2003 she was collaborated with Polish Telecom R&D Centre. She participated in several European and national projects (2004–2008). Her research interests focus on NGN and NWGN architectures, node and network virtualization, signaling system performance, admission control and resource allocation methods and queueing mechanisms. She is the author and co-author of more than 40 research papers presented in books, journals, and conference proceedings and about 30 technical reports.

e-mail: halina@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Wojciech Burakowski was born in Warsaw, Poland, in 1951. He received his M.Sc., Ph.D. and D.Sc. degrees in telecommunications from the Warsaw University of Technology in 1975, 1982 and 1992, respectively. Now he works as Full Professor at the Institute of Telecommunications, Warsaw University of Technology and the National Institute of Telecommunications, Warsaw. He leads TNT research group. Since 1990 he has been involved in many COST and EU Framework Projects. He is a member of Telecommunications Section of the Polish Academy of Sciences and an expert in 7 FR Programme. He was a chairman and a member of many technical programme committees of national and international conferences. He is the author or co-author of about 170 papers published in books, international and national journals and conference proceedings and about 70 technical reports. His research areas include new networks techniques, ATM, IP, heterogeneous networks (fixed and wireless), network architecture, traffic engineering, simulation techniques, network mechanisms and algorithms.

e-mail: wojtek@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland