Resource Control in IP Networks

Gábor Fehér

- Ph.D. Dissertation -

Supervised by

Dr. László Gefferth and Dr. István Cselényi
High Speed Networks Laboratory
Department of Telecommunications and Telematics
Budapest University of Technology and Economics

Budapest, Hungary
May, 2003
The recensions of this Ph.D. dissertation and the minutes of the public Ph.D. defense are available in the Dean’s Office.
# Table of Contents

1 **Introduction** ............................................. 1

2 **Teleservice Broker Systems** .................. 2
   2.1 Non Adaptive Teleservice Broker Systems ................................................. 2
   2.2 Teleservices Relying on Adaptive Transport Protocols ............................... 2
   2.3 An Adaptive Teleservice Broker System Proposal ...................................... 3
      2.3.1 The Main Concept of the Adaptive Teleservice Broker System .... 3
   2.4 Teleservice Description for Adaptable Teleservices .................................. 4
      2.4.1 Teleservice Definition...................................................................... 4
      2.4.2 Teleservice Configuration Validation .............................................. 7
      2.4.3 Teleservice Evaluation Expressions ................................................ 7
      2.4.4 Special Teleservice Evaluation ...................................................... 11
      2.4.5 Penalization Coefficients ............................................................... 12
   2.5 Other Teleservice Descriptions .................................................................. 12
   2.6 Teleservice Resource Reservation Procedure ............................................ 13
      2.6.1 The Black-Box Network................................................................ 13
      2.6.2 Measurements Instead of Resource Reservation ............................. 14
      2.6.3 Teleservice Resource Negotiation Procedure ................................ 14
      2.6.4 Static Negotiation Procedure ......................................................... 14
      2.6.5 Dynamic Negotiation Procedures .................................................... 16
      2.6.6 The Bottom-Up Negotiation Algorithm........................................ 17
      2.6.7 The Top-Down Negotiation Algorithm ........................................... 19

3 **Performance Analysis of Teleservice Broker Systems** .......... 21
   3.1 Performance metrics ................................................................. 21
      3.1.1 Adaptation of Classical Performance Metrics .................................... 21
      3.1.2 New performance metrics ............................................................ 23
   3.2 Performance Analysis Using Simulations ............................................. 24
      3.2.1 The Simulation Environment ........................................................ 24
      3.2.2 Simulated Teleservice Brokers ...................................................... 26
      3.2.3 Network Load .............................................................................. 27
      3.2.4 Simulation Time ........................................................................... 28
   3.3 Analysis of Bottom-Up Negotiation Algorithm Variants ............. 28
3.4 Analysis of Top-Down Negotiation Algorithm Variants...............29
3.5 Comparison of the Teleservice Initiation Techniques ..................31
  3.5.1 Average Teleservice Setup Step and Average Teleservice Adaptation Step Count Performance Metrics .................31
  3.5.2 Teleservice Blocking Probability Performance Metric.......33
  3.5.3 Teleservice Quality Distribution Performance Metric.........34
  3.5.4 Negotiation Skill Performance Metric .............................37
3.6 Analysis on Different Simulation Scenes ..................................37
  3.6.1 Simulation Using a Different Teleservice ..........................38
  3.6.2 Simulation on an Alternative Network Scenario .................41
3.7 The Required Computing Capacity of the Negotiation Algorithms ...44
3.8 Overall Conclusions ...........................................................45
3.9 The Future of Adaptive Teleservice Broker Systems .................46

4 Analysis of Routers Supporting Resource Reservation ..............47
  4.1 Resource Reservation Capable Router Model .........................48
  4.2 Resource Reservation Protocol Features ..............................48
  4.3 The Building Blocks of Resource Reservation Protocols ............52
    4.3.1 The Traffic Forwarder Block .......................................53
    4.3.2 The Signaling Message Processing Block .......................54
    4.3.3 The Session Maintenance Block .................................55
  4.4 The Router Model ..........................................................56
    4.4.1 The Validation of the Router Model .............................56
    4.4.2 Similar Router Models ............................................56
  4.5 Performance Metrics ......................................................57
    4.5.1 Average Signaling Message Processing Time ..................58
    4.5.2 Average Traffic Forwarding Time ...............................58
  4.6 Router Load Analysis ......................................................59
    4.6.1 The Load on the Traffic Forwarder Block ......................60
    4.6.2 The Load on the Signaling Message Processing Block ........61
    4.6.3 The Load on the Session Maintenance Block ...................61
    4.6.4 The Parameters of the Router Load .............................62
  4.7 Performance Characteristic Equations Based on the Router Model ..62
    4.7.1 Processing Power Scheduling in the Router Model ............62
    4.7.2 The Performance Characteristic Equation Profile of the Average Signaling Message Processing Time Metric ...64
4.7.3 The Performance Characteristic Equation Profile of the Average Traffic Forwarding Time Metric .......................................................... 65

5 Benchmarking Resource Reservation Capable Routers 67

5.1 Measurement Methodology .................................................................. 67
  5.1.1 Benchmarking Measurement Scenario ............................................. 68

5.2 Determining the Exact Performance Characteristic Equations .............. 69
  5.2.1 The Base Signaling Message Processing Time: $t_{S0}(r,m_t)$ .......... 70
  5.2.2 The Base Traffic Forwarding Time: $t_{D0}(r,s)$ .......................... 73
  5.2.3 The Signaling Message Processing Delay Caused by the Data Forwarding Task: $d_{SD}(t_{S0},t_{D0},q_{ST},t_S)$ .............................................. 76
  5.2.4 The Signaling Message Processing Delay Caused by the Session Maintenance Task: $d_{SM}(t_{S0},t_{M0},r,t_S)$ ........................................ 79
  5.2.5 The Data Forwarding Delay Caused by the Signaling Message Processing Task: $d_{DS}(t_{D0},t_{S0},q_{ST},t_D)$ ............................................ 86
  5.2.6 The Data Forwarding Delay Caused by the Session Maintenance Task: $d_{DM}(t_{D0},t_{M0},r,t_D)$ .................................................. 87
  5.2.7 Benchmarking Measurement Summary ......................................... 89

5.3 Validation of the Results ...................................................................... 91

5.4 Future Extensions to the Performance Analysis Framework ................. 92

6 Summary of the Dissertation 94

Acknowledgements 94

References 95

Publication of the New Results 99
1 Introduction

Recently, there has been a steadily increasing number of multimedia applications (e.g. IP-Telephony [1] or videoconferencing [2]) which require networks where end-to-end quality of service (QoS) [3][4][C3] is provided. The QoS provisioning is based on a contract between the user and the network operator where the operator guarantees that the service received by the users, expressed in the amount of network resources (e.g., effective bandwidth), is assured regardless of other users’ traffic in the network. Unfortunately, for many reasons, until this time there have been no solutions that offer end-to-end QoS in multi domain networks, such as the Internet [O3][O4].

In my thesis work I have focused on the possibilities of supporting QoS multimedia applications over present-day IP networks. There are two different levels at which I have proposed a solution to control the network resource demands of multimedia teleservices.

First, I have analyzed teleservice initiation techniques realizing high level resource management for teleservices. Nowadays, many standards and proposals exist in the field of teleservice session control, like the H.323 [7][8] or the Session Initiation Protocol (SIP) [10] extended with the Real-Time Transport Protocol (RTP) [11][12]. However both solutions have certain disadvantages. Systems based on the H.323 protocol cannot cope with the fluctuation of the available network resources, while those solutions that rely on the RTP protocol are restricted to handle teleservice sessions as a set of adaptive, but independent connections, thus achieving less effective teleservice management. In my dissertation I have proposed an adaptive teleservice broker system that combines the strengths of recent approaches. As a result this teleservice broker initiates services that are adapted to the given network conditions respecting the unity of the teleservice. Finally, I have concluded my investigations with the performance comparison of the proposed adaptive teleservice broker system to other teleservice initiation techniques.

Second, I have examined the current resource reservation signaling protocols realizing low level resource control. These signaling protocols differentiate network flows utilizing IntServ-based [3] or DiffServ-based [4] QoS provisioning. The number of such resource reservation protocols has also been steadily increasing in the last decade, including a hard state protocol, ST-II [29] and several soft-state protocols, like RSVP [30], DRP [31], YESSIR [32] as well as light-weight approaches, such as Boomerang [C3] and Load Control [33]. Moreover, recently there have been more and more QoS-aware IP router products supporting one of these QoS signaling protocols, mostly RSVP. However, in spite of the wide range of router products and versatility of resource reservation protocols, the deployment of dynamic, signaling based QoS provisioning is still limited in today’s Internet. One of the main reasons for this is that there is not enough information on the scalability of these signaling protocols and its impact on the overall performance of the network. Additionally, there are no standards how to compare the performance of reservation capable routers in order to design QoS enabled networks [C6]. In my dissertation I have set up a performance analysis framework that describes the performance characteristics of resource reservation capable routers in the aspect of the joint effects of traffic forwarding and signaling processing.
2 Teleservice Broker Systems

The lack of end-to-end QoS provisioning in recent networks forces the multimedia applications to run on best-effort networks or on such networks where the QoS provisioning does not cover the full path of the connections. For this reason, these multimedia services suffer from quality degradations, when the network load raises. One approach to cope with the fluctuation of the available network resources is to use adaptive protocols and applications that could adjust the resource demand of a media connection to the capacity of the bearer connection. A more sophisticated alternative can be a teleservice broker application [5][6] that coordinates the initiations of not just the connections, but the whole teleservice.

In order to control teleservice session, there are already existing standards, like the H.323 protocol family [7][8] that is an ITU standard for multimedia conferencing or a novel proposal discussed in the IETF Multi-party Multimedia Session Control (MMUSIC) working group called Session Initiation Protocol (SIP) [10]. Both of them realize multimedia teleservice initiation, plus the H.323 protocol also describes the management of the teleservices.

2.1 Non Adaptive Teleservice Broker Systems

Till these days, the most well known IP conferencing and telephony system is the H.323 recommendation that was standardized in 1996 by the Telecommunication Standardization Sector of ITU. Originally, it is a signaling protocol that is used to negotiate a videoconference between two parties in a non QoS aware, packet switched local area network. The second version of the recommendation with new extensions to the original concept makes possible to operate the protocol in corporate sized networks or even on the Internet. Now, H.323 is ready to utilize QoS enabled networks and capable of initiating multiparty teleservices also.

The H.323 system consists of H.323 capable terminals, gatekeepers, gateways and multipoint controller units. The gatekeepers, in addition to the identification of the users who participate in the service, also control the resource needs of the services. They have the power to limit the bandwidth used by the conferences and to control the traffic of the managed local area networks through the gateways. The gateways provide worldwide connectivity and interoperability for services and they are also capable to translate the media sessions into different formats. At last, the task of the multipoint controller unit is to distribute the unicast media sessions to the other participants of the conferences.

The main advantage of the H.323 protocol is that it is standardized for a long time and therefore numerous device vendors have already made their product H.323 capable. However, there are also limitations of the H.323 systems: the extreme complexity of the specification and the inherited design concept that considers local area, non QoS aware networks only. These limitations make it difficult for the H.323 systems to conquer the whole Internet [9].

2.2 Teleservices Relying on Adaptive Transport Protocols

One of the main competitors of the H.323 protocol is the Session Initiation Protocol (SIP) [10] that is an IETF standard for conference session initiation and call control. The SIP protocol, using the Session Description Protocol (SDP) [13] and the Session
Announcement Protocol (SAP) [14], provides the identification of terminals, the negotiation of media formats and the setup of services. In itself the SIP protocol is able to initiate teleservices only, but the realized teleservices are not maintained. Therefore, services, established via the SIP protocol usually apply the Real-Time Protocol (RTP) [11] to transport their media sessions. From this point of view, the SIP and RTP protocol pair provides a teleservice initiation technique that initiates and maintains teleservices.

The IETF standard RTP protocol is an adaptive transport protocol for continuous media transmission. It provides payload identification, sequence numbering, time-stamping and data transmission monitoring. Based on these functions, the multimedia applications get a feedback describing the transport capability of the utilized network links and based on this information they might decide to change the parameters of the media transfer.

However, in the case of a usual teleservice, the qualities of the different media sessions (e.g. audio and video sessions) are not independent of each other and therefore altering the transfer characteristic of a single media session might require to modify other sessions as well.

2.3 An Adaptive Teleservice Broker System Proposal

Since both investigated teleservice initiation techniques have disadvantages, I have designed a novel approach, the adaptive teleservice broker system. Utilizing the benefits of the previous techniques, the adaptive teleservice broker system is capable of reconfiguring the whole teleservice configuration according to the fluctuation of the network resources, unlike teleservice initiation techniques relying on adaptive transport protocols that adjust the transfer characteristic of individual connections only. This way, the adaptive teleservice broker systems manage and balance the quality of all the media sessions within a teleservice at the same time, achieving higher quality teleservices.

In my dissertation I have focused my research activity to the network resource requirements of the teleservice broker system. However, a fully functional teleservice broker system requires the investigations of various multimedia coders and decoders (codecs) [18][19]; the negotiation and management of the terminal capabilities and codecs [20]; and the announcement of the teleservices [14]. Since these topics are out of scope considering the IP network resource control techniques, I have assumed that the proposed adaptive teleservice broker system uses the state-of-the-art multimedia codecs, terminal capability negotiation algorithms and teleservice publication methods.

2.3.1 The Main Concept of the Adaptive Teleservice Broker System

According to the concept of the adaptive teleservice broker system, the teleservice initiation is requested by the user. He or she fills out a teleservice template offered by the Service Providers inviting other participants and requiring certain quality of media sessions among the participants. The teleservice template is sent to the teleservice broker. First, the teleservice broker validates the configuration and sends it immediately back if any of the rules that the Service Provider implemented in the service is broken. Such rules are able to express restrictions considering the number of the participants or constraints about the quality of the media sessions. In the case of a
valid teleservice configuration, the teleservice broker performs content negotiation among the participants by checking whether the participants’ terminals are ready to be involved in the teleservice. If all the terminals are present and ready then the teleservice broker initiates resource reservation requests for all the connections that are necessary to build up the desired teleservice configuration. If the resource reservation is successful then the teleservice broker launches the realized teleservice, otherwise it initiates a negotiation procedure to adapt the user’s teleservice configuration to the free network capacities. This negotiation procedure is intended to deliver an optimal teleservice configuration without any user interactions. In the optimal configuration the resource demand of the service meets the actual free network capacities and at the same time this configuration is the most preferred by the user among all the possible configurations at the same time that also fits into the network. After a successful negotiation procedure the teleservice broker launches the adapted teleservice configuration. In case of unsuccessful negotiation the requested teleservice is not realized, instead the service is blocked. During the lifetime of the teleservice the teleservice broker maintains the service by monitoring the impact of the network resource fluctuation and reconfigures the teleservice configuration when necessary. For the reconfiguration, the teleservice broker uses the same negotiation procedure as for the teleservice initiation. Finally, when the users ask to terminate the teleservice, the broker initiates release requests for all resource reservation sessions that were set up during the particular teleservice.

### 2.4 Teleservice Description for Adaptable Teleservices

The adaptive teleservice management requires some kind of intelligence for the teleservice broker application in order to negotiate the teleservice configurations efficiently. This intelligence is based on the knowledge of the teleservices including the awareness of the human preference considering different teleservice configurations. Thus, the adaptive teleservice broker relies on a teleservice description containing the teleservice definition, which defines the service in a formal way; and the teleservice evaluation expressions providing the possibility of ranking teleservice configurations. During the negotiation procedure, the teleservice broker uses this teleservice description to construct all the configurations that meet certain network resource limitations, and select the most preferred configuration based on the teleservice evaluation expression. See Figure 1.

![Figure 1 - Teleservice Description](image)

#### 2.4.1 Teleservice Definition

Teleservices can be really different but one this is always common, namely, they are all built up from participants having media session connections with each other. The
teleservice definition part of the teleservice description includes the specification of
the participants and media session connections.

Describing the participants, I have defined *party sets*: The party set is a group of users
who are associated to a certain behavior or role in the teleservice. The elements of the
party set are the users who has all the rights and limitations that the party set specifies.
For example, in a videoconference there can be *chairmen* and *members* as party sets,
where chairmen have the right to invite new members or force to quit others during
the service.

Portraying the media sessions, I have defined *media session sets*: The media session
set determines the type and characteristic of media information stream that can be
carried on a connection. If a user belongs to a certain media session set then he or she
is connected to all the other users residing in the same media session set using the
prescribed media information stream. In a realized teleservice, each media session set
is an individual multicast connection or a set of unicast connections, but different
media session sets always use different connections also. Beside the type of the media
(e.g. audio or video), the quality of the media also differentiates media session sets.

As an example, Figure 2 presents a videoconference with three participants. Here,
John and Kate are users placed in the low quality audio session set and therefore the
broker builds up a low quality audio connection between them. Kate also participates
in the high quality audio and video session sets together with Paul, so the broker
connects them using high capacity audio and video connections.

![Figure 2 - Example for a Three Party Videoconference Service](image)

Additionally, teleservices often apply informal rules expressing restrictions,
mandatory requirements or other constraints considering a particular configuration of
the teleservice. For example, in the case of the videoconference teleservice, a rule can
declare that more than one person should be involved in the service or that the audio
connection is mandatory for every participant of the teleservice. Based on the media
session sets and their elements, the users, I have formalized these rules as
mathematical expressions. In the *service rules* I have defined mathematical operators
over the media session sets and variables representing users. The rules are Boolean
expressions whose results are true or false depending on the evaluated teleservice
configuration.
This way my proposed teleservice definition consists of the parties and media session set enumerations, which are the building blocks of the teleservice. Each party set refers to a behavior that grants or denies certain activities for the user. Besides the enumeration of the party and media session sets, there are the service rules also. As an example, Figure 3 shows a teleservice definition for a videoconference service. The informal definition of this videoconference teleservice declares that there can be up to five parties in the conference. They can participate in audio or video sessions. The rules define that there must be exactly one chairman among the users, who must participate in the video session beside the audio session, in which all participants should take part.

![Figure 3 - Teleservice Definition Example](image)

The formal description defines the involved parties and the media sets in the first two rows. There are two kinds of participants: chairman and member; and there are two kinds of media sessions: audio and video. The rows following the party and media session set enumerations declare the rules for the videoconference teleservice, where the \( \# \) operator expresses the number of elements in the set. The first rule says that exactly one chairman exists in the service. The next rule limits the number of the participants allowing maximum four of them inside a single videoconference service. Rules in the last two rows declare that the chairman must participate in both the audio and video sessions and for members the audio session is mandatory.

The proposed teleservice definition is complete with the definition blocks that specify the behavior associated to the party sets and the media properties of the media sessions sets. In the case of the party set specifications, I have listed the activities that are granted or denied for a certain party. All of these activities are in relation with teleservice configuration modification. Figure 4 presents an example for the party behavior definition that regulates the participants of a videoconference. In this example only chairmen have the right to reconfigure the teleservice.

![Figure 4 - Party Behavior Example](image)

The next element of the teleservice definition is the media definition block, which specifies the physical realization format for each media session. Most of the media session types can be realized in different quality types. The proposed teleservice
2.4.2 Teleservice Configuration Validation

I have introduced the teleservice configuration validation procedure that judges whether a certain teleservice configuration is valid or not. According to my definitions, a teleservice configuration is considered to be valid only if all the rules described in the teleservice definition are true for it. Before the teleservice initiation the teleservice broker performs the validation procedure, in order to prevent users from initiating meaningless teleservice configurations. Besides, it also aids the teleservice negotiation via ruling invalid configurations out, and thus preventing the undesired resource wasting.

2.4.3 Teleservice Evaluation Expressions

Having the teleservice definition it is easy to build up all the valid teleservice configurations that a teleservice might have. However, in order to ensure that the optimal configuration has the maximal preference, I have introduced a procedure that tells the value of a teleservice configuration. The mathematical expression used for the calculation of this value is called the evaluation expression. The result of the expression is the preference score, which indicates how much the examined configuration is preferred. The preference scores of different configurations within the same teleservice can be compared to each other and this way a ranking can be established among all the configurations. The evaluation expression is constructed in a way that higher preference score represent higher preference, and for practical
reasons the preference score is weighted to be always between 0 and 1 expressing the level of the preference on a percentage scale.

Of course there are several aspects how someone prefers a configuration over other ones. Therefore, I have not limited the number of evaluation expressions specified in the teleservice description. For example, one of the evaluation expressions might express configuration ranking according to the user’s satisfaction and a second one might help to select the optimal configuration considering the price of the service and the user’s satisfaction at the same time. In this case, the user, who requests the teleservice, is able to select one of the teleservice evaluation expressions before the teleservice initiation.

There are also many ways to generate an evaluation expression and the service provider is not bounded to any restrictions, he or she can apply arbitrary expression to be used as teleservice evaluation expression. However, in the case of my teleservice broker analysis there are several performance metrics that are based on the preference score of teleservice configurations. To be fair in the analysis I have defined an evaluation expression that aims at expressing a realistic preference of the configurations. In my dissertation, I have constructed a user model that evaluates the preference score according to the joint user’s satisfaction and price aspects, since I have assumed that these are the two main aspects that an everyday user takes into account. Based on these two aspects, the higher quality the user gets the higher his or her preference is. However, due to the price of the resources, the higher resource demand the connection has the less is its preference. I have defined a preference table for each media sessions containing the preference values according to the realizable qualities. This is the media preference value. Table 1 and Table 2 show the preferences I have obtained for certain media streams:

<table>
<thead>
<tr>
<th>Resource demand [kbps]</th>
<th>Audio0</th>
<th>Audio1</th>
<th>Audio2</th>
<th>Audio3</th>
<th>Audio4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preference [%]</td>
<td>0</td>
<td>25</td>
<td>45</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource demand [kbps]</th>
<th>Video0</th>
<th>Video1</th>
<th>Video2</th>
<th>Video3</th>
<th>Video4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preference [%]</td>
<td>0</td>
<td>35</td>
<td>65</td>
<td>85</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
</tbody>
</table>

Besides the pure media preference the model considers that the users prefer teleservices where the perceived qualities of the media sessions are more balanced. For example, the users of a videoconference might dislike if the video connection has really high quality, but the audio stream bears with low quality only, instead an average quality for both media sessions is more preferred. In my proposed evaluation expression the service quality balancing is ensured by penalizing media sessions
realized on different qualities. Equation 1 shows the user preference function \( u_p \) expressing the preference of a teleservice user, who participates in several media sessions:

\[
up(i, M) = \frac{1}{m_i} \sum_{x \in M_i} mp(x) - \frac{1}{m_i \cdot (m_i - 1)} \sum_{x, y \in M_i, x \neq y} b_1 \cdot |mp(x) - mp(y)|,
\]

where \( M_i \) is the set of the media sessions, in which user \( i \) participates; \( m_i \) denotes the number of elements in the \( M_i \) set; the \( mp(x) \) function tells the preference value of media session \( x \); and \( b_1 \) is the coefficient that penalizes the unbalanced media session qualities. The first tag in the expression is the average of the media session preferences, while the second tag is the average of the weighted media session preference dissimilarities.

Finally, the whole evaluation expression, as it is shown in Equation 2, sums up the user preference functions for each user and also ensures the fairness of the teleservice punishing the dissimilarity between user preference values. This equation is similar to the previous one, but here the user preference values are accumulated, instead of the media preference values:

\[
PS(P, M) = \frac{1}{p} \sum_{x \in P} up(x, M) - \frac{1}{p \cdot (p - 1)} \sum_{x, y \in P, x \neq y} b_2 \cdot |up(x, M) - up(y, M)|
\]

where \( P \) and \( M \) are the party and media session sets representing the actual configurations; \( up(x,M) \) is the user preference function for user \( x \); \( p \) denotes the number of the participants; and finally, \( b_2 \) is the penalization weight for the dissimilar user preference values.

In order to have an evaluation expression, whose result is always between 0 and 1 the penalization constants \( b_1 \) and \( b_2 \) should be bounded. Since the shape of Equation 1 and Equation 2 are similar I have calculated the bound of Equation 2 only and then I have derived the bound of Equation 1 from the previous result.

In the case of Equation 2 I have considered that there is an \( r(x) \) ranking function that tells the place of a certain \( u_p(x,M) \) function considering the values of the \( u_p(x,M) \) functions in an descending order having different \( x \) parameters. Based on this ranking, I have defined an \( u_p'(i) \) function, where \( i \) goes from 1 to \( p \), and the result of this function is calculated as follows:

\[
up'(i) = u_p(\text{inv}(r(i)), M),
\]

where the \( x \) parameter of the \( u_p(x,M) \) function is the inverse of the \( r(x) \) function. This inverse function gives that participant whose user preference score is the \( i \)th, when the user preference scores are arranged in descending order. Thus I have gained an \( u_p'(i) \) function, where the values of \( i \) parameter are not users, but scalars.

Among the \( u_p'(i) \) functions, the value of \( u_p'(1) \) is the highest and \( u_p'(p) \) is the lowest. I have transformed Equation 2 to use the introduced \( u_p'(i) \) function:

\[
PS(P, M) = \frac{1}{p} \sum_{i=1}^{p} u_p'(i) - \frac{1}{p \cdot (p - 1)} \sum_{i=1}^{p} \sum_{j=1}^{p} b_2 \cdot |u_p'(i) - u_p'(j)| \cdot I_{i \neq j}.
\]
In order to limit Equation 3 to be between 0 and 1, the average of the user preferences should never be less than the penalization tag as Equation 4 shows:

$$\frac{1}{p} \sum_{i=1}^{p} u^p(i) \geq b_2 \cdot \frac{1}{p \cdot (p-1)} \sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j}. \quad (4)$$

In the case of the penalization tag, I have opened the summation tag in Equation 4 on the right side and got the following approximations:

$$\sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} = 2 \cdot \left( |u^p(1) - u^p(2)| + u^p(1) - u^p(3) + K + u^p(1) - u^p(p) \right) + 
\sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} \leq 2 \cdot \left( |u^p(1) - u^p(2)| + (p-3) \cdot |u^p(2)| + K + |u^p(1) - u^p(p)| \right),$$

This equation in the case of odd number of participants can be bounded as follows:

$$\sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} = 2 \cdot \left( |u^p(1) - u^p(2)| + (p-3) \cdot |u^p(2)| + K + 0 \cdot |u^p(p-1) - u^p(p)| \right) - 2 \cdot |u^p(\frac{p+1}{2}) + K - (p-1) \cdot |u^p(p)| \right),$$

while in the case of even number of participants, it is bounded as follows:

$$\sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} \leq 2 \cdot \left( |u^p(1) - u^p(2)| + (p-3) \cdot |u^p(2)| + K + 0 \cdot |u^p(p-1) - u^p(p)| \right),$$

With these two upper limits, I have expressed the penalization tag, in Equation 4. In the case of odd $p$ numbers:

$$b_2 \cdot \frac{1}{p \cdot (p-1)} \sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} \leq 2 \cdot b_2 \cdot \frac{1}{p} \cdot \left( \frac{p-1}{p-1} \cdot |u^p(1)| + \frac{p-3}{p-1} \cdot |u^p(2)| + K + \frac{0}{p-1} \cdot |u^p(\frac{p-1}{2})| \right),$$

While in the case of even $p$ numbers:

$$b_2 \cdot \frac{1}{p \cdot (p-1)} \sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} \leq 2 \cdot b_2 \cdot \frac{1}{p} \cdot \sum_{i=1}^{p} |u^p(i)|$$

$$b_2 \cdot \frac{1}{p \cdot (p-1)} \sum_{i=1}^{p} \sum_{j=1}^{p} |u^p(i) - u^p(j)| \cdot I_{i\neq j} \leq 2 \cdot b_2 \cdot \frac{1}{p} \cdot \sum_{i=1}^{p} |u^p(i)|$$
Substituting these upper bounds into the penalization tag of Equation 4, I have obtained the following relation:

\[
\frac{1}{p} \sum_{i=1}^{p} up'(i) \geq 2 \cdot b_2 \cdot \frac{1}{p} \sum_{i=1}^{p} up'(i) \\
\Rightarrow b_2 \leq \frac{1}{2}
\]

This means, that using the proposed evaluation expression, the \(b_2\) penalization coefficients cannot be larger than 0.5.

Considering Equation 1, the determination of the \(b_1\) penalization coefficient is analogue to the previously described method. As a results I have specified that the upper bound of the \(b_1\) parameter is also 0.5.

### 2.4.4 Special Teleservice Evaluation

Despite that the teleservice provider prescribes the media session realization qualities having 100% preference of the most demanding quality settings, it is also possible that the user who requests the teleservice does not require the maximal quality from certain media sessions, instead his or her preference is a less resource demanding configuration. In this case the results of the evaluation expression need correction, since the 100% preference score should always reflect the configuration that the user originally requested. For this case, I have suggested to modify the preference of the media session realization qualities. A simple way is to keep the slope of the curve and fit it to a scale of resource demands, where the maximal resource demand goes to the 100% preference. Since the resource demands of each media session realization quality is defined by the teleservice operator, they cannot be changed by the user and therefore the number of possible media session realization qualities is reduced in the service.

It is also possible, that the initiator user requests a teleservice that is intentionally unbalanced in the aspect of the media connections. In the initiator user’s concept the required teleservice is the most preferred one and therefore the evaluation expression should readjusted to reflect his or her preference. A considerable solution is to balance both the user preference function and the teleservice evaluation expression with a new tag to respect the request of the user. I have proposed to use Equation 5 instead of Equation 1:

\[
up(i, M) = \frac{1}{m_i} \sum_{x \in M_j} mp(x) - \frac{1}{m_i - (m_i - 1)} \sum_{x,y \in M_j, x \neq y} b_i \cdot \left| cm(x, y) - (mp(x) - mp(y)) \right|, \tag{5}
\]

where the new \(cm(x,y)\) function expresses the user’s choice as the difference of the medium quality realization preference between \(x\) and \(y\) media sessions.

So as with Equation 5, Equation 6 replaces Equation 2:

\[
PS(P, M) = \frac{1}{p} \sum_{x \in P} up(x, M) - \frac{1}{p \cdot (p - 1)} \sum_{x,y \in P, x \neq y} b_2 \cdot \left| cup(x, y) - (up(x, M) - up(y, M)) \right|, \tag{6}
\]

where the new \(cup(x,y)\) function expresses the user’s choice reflecting the difference between the user preference function in the case of user \(x\) and \(y\).
2.4.5 Penalization Coefficients

The proposed teleservice evaluation expression is based on a user model that imitates the human behavior preferring well balanced teleservices. However it also contains two penalization parameters that enable to precisely customize the expression to the preference of the teleservice users. These penalization coefficients can be determined by interviewing the users of the teleservices and ask them to rank some of the teleservices. Since the number of the possible teleservice configuration combinations can be huge, it is impossible to rank all the configurations. Instead, the users are questioned about how they tolerate the unbalanced media session qualities considering one participant and how balanced should the overall quality of the media sessions be taking into account all the participants. Based on the responses the penalization coefficients can be determined.

2.5 Other Teleservice Descriptions

There are other publications proposing methods how to define the teleservices. Terminals using the H.323 protocol must also understand the H.245 Media Control Protocol [17] that defines the multimedia connections between the participants. Using the H.245 protocol, the participating terminals exchange their decoder capability set (bit rates and various options for video and audio codecs that can be accepted by the decoder), and also indicate the maximum number of simultaneous audio and video streams they are capable of decoding. The media capabilities are described as a set of capability descriptors, listed in decreasing order of preference. A capability descriptor is a set of alternative capability sets, where each alternative capability set contains a list of algorithms, only one of which can be used at any given time. Beside the capability negotiation, H.245 provides conference floor control [16] and manages the setup and tear down of media channels as well.

A more recent teleservice description is the Session Description Protocol (SDP) [13] proposed by the IETF. The purpose of SDP is to convey information about media streams in multimedia sessions to allow the recipients of a session description to participate in the session. The multimedia session, within the SDP context, is defined as a set of media streams that exist for some duration of time. Using SDP the teleservice initiator defines properly all the sources, formats and timing of the multimedia connections in a textual form. The session description includes:

1. Session name and purpose
2. Time, when the session is active: A list of start and stop times bounding the session.
3. The media comprising the session: The type of media (video, audio, etc); the transport protocol (RTP/UDP/IP, H.320, etc); the format of the media (H.261 video, MPEG video, etc); the source address and transport port for media.
4. Information about the bandwidth to be used by the conference (optional)
5. Contact information for the person responsible for the session (optional)

The SDP session descriptions are entirely textual opposed to a binary encoding such as H.323. The textual form enhances portability, enables a variety of transports to be used (e.g. e-mail messages) and allows flexible, text-based toolkits to be used to generate and to process session descriptions SDP is intended to be general purpose so that it can be used for a wider range of network environments and applications than
just multicast session directories. However, it is not intended to support negotiation of session content or media encodings - this is viewed as outside the scope of session description.

Because SDP addresses the description of teleservices only, the IETF MMUSIC working group proposes also a new protocol, called Simple Conference Control Protocol (SCCP) [15] to be used for tightly coupled conferences. In addition to the SDP protocol, SCCP services provide floor control to implement access control rules for distributed application resources, manage the set of application/media sessions and also support functionality for management of participants. The three main services of the SCCP protocol are:

1. Conference Management: managing a conference context containing membership information of all current conference participants. (invite others, join/leave/terminate conferences)
2. Application Session Management: controlling a set of media/application sessions that constitute the conference. (negotiating and changing the configuration, join/leave/create/terminate sessions)
3. Floor Control: supporting application state synchronization. (grab/inhibit/release/test/ask/give floor)

Compared to the present teleservice descriptions, the novelties of my teleservice description proposal are: (i) the flexibility of the media and participant associations based on mathematical rules; and (ii) the ability to express the value of a certain configuration.

### 2.6 Teleservice Resource Reservation Procedure

In my proposed adaptive teleservice broker system, the broker application manages the resource reservation for all the connections of the teleservices. Nevertheless, it does not necessarily mean that the broker controls the network resources, instead the broker utilizes resource reservation protocols initiating the signaling of the resource reservations. The resource reservation attempts initiated by the teleservice broker might fail due to network resource shortage. In this case the teleservice broker opens a negotiation procedure and adapts the teleservice configuration to an optimal configuration. The optimal configuration is the particular configuration that network resource demand does not exceed the available network resources and among all the realizable configurations this one has the highest preference score according to the evaluation aspect that the user has chosen.

#### 2.6.1 The Black-Box Network

During the negotiation procedure I assume that the adaptive teleservice broker application does not have any information about the network where it manages the teleservices. This assumption is essential, since I have designed my adaptive teleservice broker system to work on the Internet that is a huge set of domains provided by different network operators. Due to strategic reasons network operators do not publish relevant information about their networks, and therefore the adaptive teleservice broker does not know the network initially. Nowadays many applications exist that discover the topology of the unknown networks [21][22]. Since, these applications work on the IP layer, thus they are not able to detect the perfect resource conditions of the discovered network. The lower layer IP tunneling [23], MPLS [24]
or other kinds of traffic engineering [25] or even a shared medium, such as the Ethernet technology, all bias the correctness of the detected network topology and its resources. For this reason, the proposed adaptive teleservice broker is designed such that it does not require any information about the network, where it provides services, but instead it performs an iterative negotiation, guessing the available resources for its teleservices.

2.6.2 Measurements Instead of Resource Reservation
In networks that support QoS protocols, the resource reservation is performed using signaling messages that instruct the routers of the network to dedicate resources for the specific teleservice session. However, currently the majority of the Internet domains do not support any network resource allocation or traffic flow differentiation. Here, the teleservice broker cannot protect the teleservices from other traffic, but still able to receive feedbacks about the quality of the connections (e.g. end-point send RTCP messages or active measurements) and discover if the resource demand of a service meets the available network resources. When the active or passive measurements result that the network is able to provide the requested QoS parameters of the service then the service is considered to be reserved. Since in this case the network resources are not dedicated to the teleservice sessions, the traffic of other applications might also load the network so severely that the quality of the teleservice can no longer be maintained. For this reason, in my proposal, when a teleservice notices quality falls over a connection then it notifies the teleservice broker that initiates the reconfiguration of the teleservice. Similarly, when the network load goes down, the teleservice broker reconfigures the downgraded teleservices to raise their quality.

2.6.3 Teleservice Resource Negotiation Procedure
The teleservice resource negotiation takes place, when the teleservice resource reservation procedure fails to reserve the network resources that are essential to build up the teleservice that a user requested. The mission of the negotiation is to degrade the teleservice configuration in a way to achieve the optimal configuration. However, usually due to time constraints it is not feasible to test all the possible configurations and find out all the realizable teleservice configurations. This way only a part of the teleservice configurations can be checked and the most preferred configuration is selected from this set only. However, this method cannot assure that the adapted configuration is truly the optimal one. Therefore, I have applied a relaxation for the mission of negotiation procedure and adjusted its goal to deliver a near optimal configuration that is close to the optimal configuration, but is not necessarily the same.

2.6.4 Static Negotiation Procedure
The static negotiation procedure [C2] is an early proposal of Károly Farkas and me to command the negotiation during the teleservice resource reservation procedure. This negotiation procedure is an iterative procedure, where the configuration chosen for the next reservation attempt is selected according to the success or failure of the previous teleservice configuration resource reservation attempt. Thus, a sequence of teleservice configurations is allocated successfully or unsuccessfully until the procedure meets the near optimal configuration. In our proposal the designer of the teleservice
predefines a static negotiation graph indicating the negotiation sequences that guide the negotiation procedure. In the negotiation graph the nodes are different teleservice configurations and the transitions show the possible next attempts of the negotiation sequence. If the resource reservation is successful then the state of the negotiation forwards to the node that was tested and from this time the broker investigates the realization of those configurations that are available from that node. When there is more than one transitions starting from a certain node then the sequence of the attempts is determined randomly. The walking along the transitions is repeated until the broker gets to a node of the graph, from which all the transitions refer to configurations that are unrealizable in the actual network conditions. The result of the negotiation is the configuration, where the broker stands in the negotiation graph at that time. The negotiation procedure is called “static”, since the service provider should design the whole negotiation procedure and negotiation graph in advance.

As an example Figure 6 shows one possible negotiation graph for an audio conference teleservice, where there are three participants and only one medium, the voice session. In the figure, the participants are denoted by black bubbles and the transitions are symbolized by red arrows. A black line shows that two audio conference members participate in the voice session and have connections between each other. The upper position, where there is an empty configuration, is the starting position of the static negotiation procedure.

![Figure 6 - Static Negotiation Graph Example](image)

The drawback of the static negotiation procedure is that the service provider has to define the full negotiation graph and its transitions at the service design time. The more node the negotiation graph has the more transition is associated with the graph and this makes the teleservice definition larger. I have investigated how the size of the negotiation graph grows as the number of participants and medium instances increases in the teleservice:

I have considered a teleservice where the maximum number of allowed participants is $p$ and there are $m$ medium instances having $k_i$ number of different quality realizations, where $i$ refers to the medium. This time, I have assumed no rules in the service, the participants might participate as many media sessions as they like. Using this
notation, the number of different medium connection possibilities between two participants \((k)\), is calculated as follows:

\[ k = (k_1 + 1) \cdot (k_2 + 1) \cdot \ldots \cdot (k_m + 1), \]

where \(k\) also includes those situations, where some of the media sessions are not presented in the connection. I have also calculated the number of possible connections between the participants \((c)\) in the whole teleservice that is:

\[ c = \frac{p \cdot (p-1)}{2}. \]

Since each media connection can be realized in \(k\) ways, the number of the different teleservice configurations is:

\[ k^c = \left(\left(\left(\left(k_1 + 1\right) \cdot \left(k_2 + 1\right) \cdot \ldots \cdot \left(k_m + 1\right)\right) \right)^\frac{p(p-1)}{2}\right). \]

This way I have shown that the number of the different teleservice configurations grows exponentially along the number of participants and the number of the different medium realization qualities also seriously affects the configuration number.

For example, an average teleservice where there are maximum 4 participants, and there are three media having 3 quality realizations each results \(64^6 = 68,719,476,736\) different configurations. In this case the service designer has to implement transitions for over 68 billion different negotiation graph nodes and these configurations have to be stored also that might exceed several terabyte capacity. Of course, there might be rules and restrictions regarding to the configurations in the teleservice that limit the number of stored configurations and transitions, but the required storage space is still unreasonably high.

On the other hand, for mini teleservices, like the audio conference service for three participants mentioned in the previous example the static negotiation procedure might be fruitful. These mini teleservices require only a small amount of storage space and the declaration of the transitions is also easy.

### 2.6.5 Dynamic Negotiation Procedures

Since the proposed static negotiation procedure is efficient only for mini teleservices, the negotiation procedure for larger teleservices needs a novel approach. To overcome the limitation of the static negotiation procedure, I have suggested the dynamic negotiation procedure, where there are no stored teleservice configurations, all the tested configurations are constructed at the runtime of the procedure.

The proposed dynamic negotiation procedure is driven by a dynamic negotiation algorithm that (i) constructs a limited set of configurations using the teleservice description; (ii) from this configuration set the negotiation procedure tests some; (iii) based on the results of the reservation attempts the negotiation algorithm stops and yields the near optimal configuration or constructs a new set of configurations and repeats the testing. I have designed two negotiation algorithms:
2.6.6 The Bottom-Up Negotiation Algorithm

First, I introduce the *bottom-up* dynamic negotiation algorithm. The basic idea of the algorithm is to increase the network resource demand of the teleservice align with the preference of the configuration from zero until the resource demand meets the actual network resource limitations. The zero network resource demand is the empty configuration that is the initiation point of the negotiation, while the target is the near optimal configuration. In between, each new negotiation step is constructed by extending the configuration tested in the previous step. This extension happens by upgrading some of the existing connections to a higher quality realization that naturally increase the network resource demand of the configuration at the same time. In the following, the configuration that is to be extended is called the *base configuration*, while those configurations that are already extended compose the *extended configuration set*. The configurations of the extended configuration set are validated against the rules of the teleservice description and invalid configurations are eliminated from the set. In the case of the bottom-up algorithm the next negotiation step is a resource reservation test of the teleservice configuration from the extended configuration set, which has the highest preference score according to the applied configuration evaluation aspect. After selecting the configuration, the negotiation procedure tries to reserve network resources for the teleservice configuration and if it is successful then the negotiation continues by constructing a new extended configuration set based on the successfully reserved configuration. However, if the configuration reservation fails then the tested configuration is removed from the extended configuration set. For practical reasons all configurations that have higher resource demands than the previously failed configuration has, are also eliminated from the extended configuration set. If there are no more configurations in the extended configuration set then it means that the actual configuration cannot be improved, so the near optimal configuration is the actual base configuration. However, in that case when the extended configuration still contains configurations after the reduction then the negotiation is repeated from that phase where the configuration with the highest preference score is selected to be reserved. The flowchart of the bottom-up algorithm can be seen in Figure 7.
In the case of the bottom-up algorithm, the construction of the extended configuration set provides some freedom to the implementation. Besides, it is also expected that the behavior of the algorithm seriously depends on the size of the extended configuration set. In the case when the extended configuration set is too large then the teleservice broker might waste significant amount of time testing configurations until it finds the one with such resource needs that can be satisfied. On the other hand, when the extended configuration set is too small then the teleservice broker might test configurations successfully but unnecessary, since using a larger extended configuration set would reduce the number of reservation attempts. This indicates that the size of the extended configuration set that can be adjusted in the implementation, affects the number of negotiation steps that the algorithm takes in order to achieve the near optimal configuration. To specify the size of the extended configuration set, I have introduced the configuration distance parameter:

**Configuration Distance**

The value of the configuration distance parameter expresses how far two configurations are from each other. The distance is defined as the number of media sessions, in which the compared configurations realize different qualities:

\[
CD(sc_1, sc_2) = \sum_{m \in M} \sum_{c \in C(m)} I_{rl(sc_1,m,c) \neq rl(sc_2,m,c)},
\]

where the \( rl \) function provides the realization quality for teleservices \( sc_1 \) and \( sc_2 \), in the case of the \( m \) medium on the \( c \) connection. \( M \) denotes the media set and \( C(m) \) expresses the set of those connections, on which medium \( m \) is realized.

In my bottom-up algorithm implementation, during the construction of the extended configuration set, I have built up only those configurations, which configuration distance compared to the base configuration was below of a given limit. This construction method provides a tool to adjust the size of the extended configuration set by adjusting the configuration distance parameter.
Negotiation Step Limit

In addition to adjusting the size of the extended configuration set, it is also possible that the negotiation algorithm performs too many negotiation steps. Since the number of the negotiation steps affects the setup time of the teleservice, I have introduced the negotiation step limit parameter that bounds the number of the negotiation steps. When the number of the negotiation steps exceeds the negotiation step limit then the negotiation algorithm stops and the delivered configuration is the best achieved configuration so far. In the case of the bottom-up algorithm, when the negotiation step limit stops the negotiation then the result of the negotiation is the last base configuration. Applying the negotiation step limit makes the teleservice setup faster, but as a tradeoff, the adapted configurations are not so close to the optimal configuration than configurations adapted by an unbounded bottom-up algorithm.

Teleservices with different number of participants and media sessions are constructed in different ways during the teleservice negotiation procedure. For this reason they might also differ in their most effective configuration distance and negotiation step limit parameters. In order to obtain these parameters for a certain teleservice, I suggest performing simulations.

2.6.7 The Top-Down Negotiation Algorithm

The second algorithm is the opposite of the bottom-up algorithm in some manner. Here, during the negotiation procedure the teleservice broker makes efforts to reserve teleservice configurations in which the network resource demand is decreased until the reservation becomes successful. The negotiation procedure is initiated with the configuration that the user requested, called base configuration analogue to the bottom-up algorithm. The negotiation algorithm is initiated because the reservation of the user requested configuration failed, therefore the algorithm constructs the degraded configuration set containing those teleservice configurations, in which the medium realization qualities along the connections are degraded compared to the base configuration. According to the lower qualities, each element of the degraded configuration set has lower resource demand than the base configuration and that is why these configurations have better chance to be successfully reserved in the network. From the degraded configuration set, the algorithm selects the configuration that has the highest preference score and tries to reserve network resources for it. In case of success, this is the near optimal configuration. In case of reservation failure, the tested configuration is eliminated from the degraded configuration set and the negotiation algorithm is repeated from the selection of the configuration with the highest preference score. Moreover, in this case, not only the tested configuration, but all the configurations that have higher resource demands than the tested one are removed from the degraded configuration set, since they surely have no chance to fit into the network. When there are no configurations left in the degraded configuration set then the negotiation algorithm steps back to the construction of a new degraded configuration set based on the last failed configuration. The flowchart of the top-down algorithm can be seen in Figure 8.
Start with the configuration that the user requested

Create the degraded configuration set by downgrading the actual connections where it is necessary

Select the configuration with the highest preference score

Try to allocate network resources for the configuration

Success

Failure

Reduce the extended configuration set by deleting unrealizable configurations

No more configurations

The actual configuration is the near optimal configuration

Figure 8 - The Top-Down Negotiation Algorithm

Similarly to the bottom-up negotiation algorithm, it is also expected that the size of the degraded configuration set seriously influence the performance of the teleservice broker. For this reason, the size of the degraded configuration set should be determined carefully. However, here the degraded set construction algorithm produces all the possible degraded configurations up to a maximal configuration distance parameter and tests them one by one in descending preference score order until one of them is successfully reserved. Even in the case of smaller configurations, the number of the reservation test could be extremely much if the teleservice configuration requires great degradation. Consequently, I propose to use a degraded configuration set construction algorithm that also forces at least a minimal degradation compared to the base configuration, through a minimal configuration distance parameter. As a tradeoff, since this construction algorithm skips some possibly good configurations, I expect that this negotiation algorithm achieves configurations that are not so close to the optimal configuration as they would be if the bottom-up negotiation algorithm would provide them.

Similarly to the bottom-up algorithm, I have suggested to limit the number of negotiation steps of the top-down algorithm. However, the top-down algorithm results a valid configuration only at the end of the algorithm, and therefore when the negotiation is stopped there are no successfully reserved configurations yet that would cause the blocking of the service. In order to avoid the teleservice blocking in this case, I have modified the top-down negotiation algorithm to check the minimal configuration first. The minimal configuration is the valid configuration that has the smallest possible resource demand among all the configurations. Checking this configuration is a great advantage in the case of the top-down algorithm. First, if the reservation of the minimal configuration fails, then there is no use to run the top-down negotiation algorithm, instead the teleservice can be blocked immediately. Second, if the reservation of the minimal configuration is successful, but the top-down negotiation bumps into the negotiation step limit then the algorithm is still able to deliver the minimal configuration as the result of the configuration adaptation.
3 Performance Analysis of Teleservice Broker Systems

In addition to the design of the teleservice broker system I have also implemented it in a simulator to validate the system and also to analyze and compare its performance to other teleservice initiation techniques. Since both the bottom-up and the top-down negotiation algorithms have certain parameters influencing the performance of the negotiation procedure I have also examined the algorithms in the aspect of these parameters.

In order to analyze teleservice initiation approaches, first I have selected performance metrics that are suitable for the characterization:

3.1 Performance metrics

Since teleservice broker systems inherit some feature of the classical network services, I have looked around and examined those traditional performance metrics that analyze the performance of the circuit switched telephone networks.

3.1.1 Adaptation of Classical Performance Metrics

I have observed that some of the classical performance metrics describing circuit switched telephone networks are also suitable to characterize the performance of a teleservice broker system with a minimal adaptation. From these metrics I have proposed to use the following ones:

Average Teleservice Setup Step Count

The average teleservice setup step count performance metric is the adaptation of the average call setup time metric used to characterize the duration of the call setup in circuit switched telephone networks. The average call setup time expresses the average of the time intervals elapsed from the request of the call till the end of the call setup. In my adaptation I have used this performance metric to characterize the length of the teleservice setup. Examining the teleservice setup procedures I have identified that the time required to setup a teleservice depends on (i) the number of teleservice reservation attempts that the teleservice broker performs in order to adapt the teleservice configuration; (ii) the length of the negotiation sequence calculation in the case of adaptive teleservice brokers; and (iii) the time that a teleservice reservation attempt takes. Considering static or dynamic negotiation algorithms, the length of the next negotiation sequence calculation is negligible compared to the effects of the other two factors. For this reason I have not considered this particular delay in the description of the teleservice setup time. In view of the length of the configuration reservation attempt I have no information about this procedure because the time that one resource reservation test takes seriously depends on the resource reservation technique and protocol that the network provides. In order to avoid the bias that resource reservation approaches might cause, I have supposed that the average duration of the resource reservation attempts is the same for similar teleservices and thus I have not represented this delay in the metric. Consequently, the length of the
teleservice setup is characterized by the average number of reservation attempts that the teleservice reservation procedure performs and therefore it is called average teleservice setup step count performance metric. The metric is calculated as follows:

\[ S = \frac{1}{|C|} \sum_{i \in C} \text{step}(i), \]

where \( C \) is the set of the teleservices that were requested during the analysis and the \( \text{step}(i) \) function tells the number of reservation attempts that was required to setup or block teleservice \( i \).

In the case of non-adaptive teleservice broker systems the number of reservation attempts are always one, since there is only one resource reservation test determining whether the service fits into the network or not. In this case the average teleservice setup step count performance metric always shows one step. However, in the case of teleservice initiation techniques that rely on adaptive transport protocols, the service is created instantly and the quality of the media session connections are adjusted later, hence in this case there are no reservation attempts. The average teleservice setup step count metric does not characterize these systems.

**Average Teleservice Adaptation Step Count**

The average teleservice setup step count metric measures the speed of the teleservice setup. However, the metric considers adapted and non-adapted teleservice initiations as well, therefore it tells nothing about the speed of the negotiation algorithms. Thus, I have defined the *average teleservice adaptation step count* performance metric to describe the duration of the bare teleservice adaptation. The metric is defined similarly to the average teleservice setup step count metric, but it takes into account only the teleservice setups where the delivered configuration was adapted by the teleservice negotiation algorithm. This way this metric describes the true speed of the negotiation algorithms:

\[ S_A = \frac{1}{|C_A|} \sum_{i \in C_A} \text{step}(i), \]

where \( C_A \) is the set of the adapted teleservices that were initiated during the examination and \( \text{step}(i) \) tells the number of resource reservation attempts that was performed during the setup of teleservice \( i \).

This metric is suitable for adaptive teleservice broker systems only, since currently this is the teleservice initiation technique that applies negotiation algorithms to manage the teleservice resource reservation.

**Teleservice Blocking Probability**

The second adapted metric is the teleservice blocking probability after the *call blocking probability* metric that characterizes circuit switched telephone networks. Initially the call blocking probability describes the ratio of the blocked calls compared to all the requested calls. In my adaptation, there are teleservices instead of calls and a teleservice is considered to be blocked if any of its media connections cannot be realized on a minimal quality level due to network resource shortage. The teleservice
blocking probability shows the ratio of the blocked and the total number of requested teleservices:

\[ P = \frac{\|B\|}{\|C\|}, \]

where \( C \) denotes the entire set of the teleservices and \( B \), which is a subset of \( C \), represents the set of the blocked teleservices.

In the case of non-adaptive teleservice brokers a teleservice setup can be successful or blocked depending on the resource demand of the requested teleservice configuration and the actual network load. However, adaptive teleservice initiation techniques have the ability to downgrade teleservices if their resource demands exceed the amount of resources that the network is able to provide. In the case of successful teleservice adaptation the teleservice blocking is avoided, which increases the number of successfully realized teleservices and reduces the teleservice blocking probability at the same time. Comparing the teleservice blocking performance metric in the case of non-adaptive and adaptive teleservice initiation techniques I have expected that it points out one of the advantages of the adaptive teleservice initiation approaches.

### 3.1.2 New performance metrics

All the adapted classical performance metrics treat the teleservice as a call and do not distinguish based on the different qualities of the realized teleservices. In order to investigate adaptive teleservice initiation approaches I have introduced new performance metrics that emphasize the differences in the delivered teleservices.

**Teleservice Quality Distribution**

I have constructed the *teleservice quality distribution* performance metric in order to measure the dominance of teleservices whose quality falls into a certain quality region. This metric represents the histogram of the teleservices realized in different qualities due to the teleservice adaptation. The dominance of teleservices in a certain quality region \( (W_i) \) is defined as the number of the teleservice configurations that belongs to the specified teleservice quality range divided by number of all the realized and unrealized teleservices:

\[ W_i = \frac{1}{\|C\|} \sum_{j \in C} I_{\{\text{score}(j) \in Q_i\}}, \]

where \( i \) determines the quality range, \( C \) is the set of the teleservices and the investigated quality range is described with the \( Q \) interval. The \( \text{score}(j) \) function tells the preference score of teleservice \( j \) that is considered as the quality of that particular configuration. In the case of unrealized teleservice configurations I have defined zero preference score.

The teleservice quality distribution metric is described as a vector of the quality region dominances, where the entire quality scale is divided into several bordering regions:

\[ W = (W_1, W_2, \ldots, W_n), \]
where \( n \) is the number of quality regions that divides the entire quality scale. The quality regions must be disjunctive sets and the union of the ranges must be the whole quality domain.

Even in the case of identical network situations, different negotiation algorithms produce teleservices of different qualities as a result of the teleservice adaptation. Thus, I have expected that they also generate different teleservice quality distribution metric. This metric stresses which quality regions become dominant as the results of the adaptation.

This performance metric is really valuable for service providers who are motivated to offer high quality teleservices. Defining the quality ranges according to the pricing of the different qualities within a teleservice, the revenue of the teleservice can be maximized.

**Negotiation Skill**

The negotiation skill performance metric shows how close the adapted teleservice configurations are to the optimal configuration. Since negotiation algorithms only deliver near optimal configuration, it is not assured that the adapted teleservice configuration has the highest score among all the feasible configurations. This metric measures the dissimilarity between the preference score of the near optimal configuration and the preference score of the optimal configuration achieved in the case of identical network conditions:

\[
NS = \frac{1}{\|C_A\|} \sum_{i \in C_A} \frac{\text{score}(i)}{\text{maxscore}(i)},
\]

where \( C_A \) denotes the set of teleservice configurations that were adapted by one of the negotiation algorithms, the \( \text{score}(i) \) function is the previously introduced function telling the quality of teleservice \( i \) and the \( \text{maxscore}(i) \) function represents the function that gives the preference score of the optimal configuration for the same teleservice request as \( i \).

The simplest method to identify the optimal configuration is to test each teleservice configurations and select the one with a successful resource reservation that has the highest preference score. Following this method, the preference score of the optimal configuration is always achievable.

### 3.2 Performance Analysis Using Simulations

In order to evaluate the performance of teleservice initiation techniques I have decided to perform simulations [O5]. Simulation is an effective performance analysis tool, when the measurements would be too slow and expensive due to the size of the investigated network. To enhance the accuracy and applicability of my results, I have designed my simulations to describe realistic environments.

#### 3.2.1 The Simulation Environment

In the case of the first simulation scene, I have used the network topology shown in Figure 9. This network scenario follows the topology of the TEN-155 network [26] as of June 1999 and thus it is highly realistic. In this network the 16 main nodes (Amsterdam, Athena, Berlin, Budapest, Dublin, Lisbon, London, Madrid, Paris,
Prague, Rome, Stockholm, Vienna, Warsaw, Zagreb and Zurich) and the connections and their capacities, as indicated on the figure, are identical to the TEN-155 network. During the simulations I have presumed that the resources in the local area networks around the capitals of the European countries are tight enough to satisfy the resource demands of the international teleservices. In this case the bottleneck links are only the backbone links that connect the capitals to each other and therefore, I have simulated no local area network activities. I have assumed that the simulated network accommodates a huge number of terminals initiating teleservices during the simulations. Here, I have presumed that in the real life the number of teleservice requests coming from a country depends on the population of the given country. This way I have distributed the terminals according to the populations of the countries. In total, there were 8354 terminals in the simulation.

In the case of the first simulation scenario I have set up the terminals to request a realistic, 5 party videoconference teleservice. This teleservice has an average resource demand. The 5 participants of the teleservice were selected randomly.

The teleservice description of the videoconference service is the same as the description of the videoconference service introduced in section 2.4, except that all participants must take part in both the audio and video sessions at the same time. The preference score of the teleservice configurations were determined in the aspect of the joint user’s satisfaction and price factors using the evaluation expression appeared in 2.4.3 with $b_1=0.33$ and $b_2=0.33$ penalization coefficients. These coefficients are based on the results of a mini-interview.
The timing of the service requests have been modeled as Poisson processes [28]: In this model, first the terminal is inactive for a certain time period that is determined using an exponential random value. After this inactive period the terminal wakes up and requests a teleservice. When the initiation request is successful, then the terminal remains active for a time period that is based on an exponential random value. At the end of the active period the terminal cancels the teleservice and falls back to inactive state. However, in the case when the teleservice initiation is blocked, the terminal waits some time and retries the initiation request for three times or until the teleservice initiation is successful. The expected value of this waiting time is also calculated using a random value with an exponential distribution. In the case of the simulated terminals, Table 3 shows the expected values of the random timing applied to initiate teleservices.

### Table 3 - Terminal Timing Parameters

<table>
<thead>
<tr>
<th>Inactive time</th>
<th>Active time</th>
<th>Waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>60 sec</td>
<td>10 sec</td>
<td>5 sec</td>
</tr>
</tbody>
</table>

### 3.2.2 Simulated Teleservice Brokers

Using simulations I have compared the performance of teleservice initiation techniques. However, in the simulator I have implemented the adaptive teleservice broker only. The other two teleservice initiation approaches were emulated by using special negotiation algorithms. I have tested the following four teleservice initiation techniques:

- **Non-adaptive teleservice broker systems**: To analyze non-adaptive teleservice broker systems, such as H.323, I have applied a *null negotiation algorithm* for the resource reservation procedure of the implemented adaptive teleservice broker system. The null negotiation algorithm has provided no mechanism to adjust the teleservice configuration to the actual network bearer capabilities imitating the behavior of non-adaptive teleservice brokers. Thus, using the null negotiation algorithm the adaptive teleservice broker has performed no adaptation when the first teleservice reservation attempt failed, instead simply blocked the teleservice.

- **Teleservices that rely on adaptive transport protocols**: To analyze the performance of teleservices that rely on adaptive transport protocols, such as SIP + RTP, I have implemented the *adaptive connections algorithm* for the adaptive teleservice broker. In this case, during the resource reservation procedure the adaptive connections algorithm increases the resource demands of each connection in the teleservice as much as possible regardless of other connections. This way it emulates the initiation of teleservices that use adaptive transport protocols in order to adjust the quality of their media connections to the network conditions. Here, unlike the bottom-up and top-down negotiation algorithms the resource demand growths of the connections are not coordinated by the teleservice broker.

- **Adaptive teleservice broker with bottom-up negotiation algorithm**: I have implemented the bottom-up algorithm for the simulation. Assuming that the size of the extended configuration sets and the negotiation step limit
parameters affect the performance of the negotiation algorithm, I have constructed several variants from this algorithm applying different values for these parameters. Based on the performance of these algorithm variants evaluated by preliminary simulations I have selected the most efficient algorithm to compare it with other approaches.

Adaptive teleservice broker with top-down negotiation algorithm: Similarly to the bottom-up algorithm I have implemented several variants of the top-down negotiation algorithm to be investigated. Based on the performance of the variants I have picked one that I have compared to the others.

Besides the negotiation algorithms of the adaptive teleservice broker system, I have implemented one more special negotiation algorithm, the brute-force negotiation algorithm.

The brute force negotiation algorithm is one possible negotiation algorithm that provides the truly optimal teleservice configuration. Using the brute force algorithm it is assured that the adapted configuration fits to the network conditions and has the highest preference among all the other suitable configurations. The brute force negotiation algorithm, according to its name, checks all the valid teleservice configurations against the free capacities in the network and determines which configurations are realizable in the case of the actual network conditions. After the testing the algorithm selects the configuration that has the highest preference score and reserves resources for it. Due to the huge number of reservation attempts, I do not recommend this algorithm to be used as a real negotiation algorithm, since it would be too slow. However, using the brute force algorithm in the simulations, I have obtained the optimal configuration that is needed for the evaluation of the negotiation skill performance metric.

3.2.3 Network Load

During the simulations, I have considered that the teleservices are not alone in the network, but other applications, such as file sharing or web browsing, also raise the network load. To describe the effect of the load of other applications, I have introduced the network load parameter that tells how many percent of the link capacities are allocated for the traffic that is not generated by teleservices. Consequently, the network load parameter also determines how much resource the network leave for teleservices. The higher the network load the less resource remains for teleservices and therefore I expect more teleservice adaptations and blockings. Unlike the fluctuating load in the network generated by freshly initiated and disappearing teleservices, I have assumed that the network load parameter is permanent during the whole simulations.

In the simulations I have investigated those situations where some parts of the network were fully utilized and therefore it caused congestions leading to teleservice adaptation or blocking. In a fully utilized link, the network load expresses the amount of resources that other applications allocate and all the rest is allocated for teleservices.
3.2.4 Simulation Time
At the beginning of the simulations I have initialized the simulator without any active teleservices, but I have preset the link capacities according to the network load parameter. At the initial transient period new teleservices experience no bottlenecks. After a while the more and more allocated teleservices cause severe congestions in the network and this situation does not change during the simulations. Because of this transient behavior biases the simulation results I have started my observation of the teleservice broker in the steady state of the simulations only.

Throughout the analysis of teleservice initiation approaches, each simulation ran for 4 simulated hours, but the teleservice requests and delivered teleservice configurations were logged only in the last 2 simulated hours. During this interval I have observed more than 40000 teleservice configurations in each simulation. All my conclusions are based on these simulation results.

3.3 Analysis of Bottom-Up Negotiation Algorithm Variants
During the description of the teleservice negotiation algorithms I have mentioned that the configuration distance and the negotiation step limit parameters are expected to affect the performance of the algorithm. In order to specify the parameter set that ensures the most effective negotiation, I have performed simulations investigating different negotiation algorithm variants using various parameters. I have used the same simulation scene for these simulations as for the comparison.

Performing first round simulations, I have determined that, in the case of the bottom-up negotiation algorithm, three variants of the original algorithm should be analyzed more thoroughly. The investigated algorithm variants have differed in their extended configuration set size limit that is controlled through the configuration distance parameter. The maximum configuration distance parameters in the three variants are one, two and three. Regarding to all the other parameters the variants were identical. Based on the preliminary simulation results, I have not limited the number of the negotiation steps in these variants.

Characterizing the performance of these three variants I have revealed that the investigated bottom-up algorithm variants, with the exception of the average teleservice setup step count and the average teleservice adaptation step count metrics, produce almost identical performance.

Average Teleservice Setup Step and Average Teleservice Adaptation Step Count Performance Metrics for Bottom-Up Variants
Figure 10 and Figure 11 present the average teleservice setup step count and average teleservice adaptation step count metrics, in which the examined bottom-up negotiation algorithm variants have differed. Both metrics are represented in the function of the network load. In the figures, the numbers after the name of the algorithm variants denote the value of the maximum configuration distance parameter.
Based on the simulation results, I can state that the performance characteristics of the different bottom-up negotiation algorithm variants are very similar to each other. Still, the fastest bottom-up negotiation algorithm, in the case of a 5 party videoconference service, is the algorithm variant that has a maximum configuration distance parameter of 2. This bottom-up negotiation algorithm variant never takes more than six reservation attempts in average and the length of average adaptation is always around 22 steps independently of the network load.

All the other investigated performance characteristics were identical in the case of the analyzed variants and therefore, I have chosen the bottom-up negotiation algorithm variant with a configuration distance parameter of 2 for the further comparisons.

### 3.4 Analysis of Top-Down Negotiation Algorithm Variants

So as with the analysis of the bottom-up negotiation algorithm variants, I have constructed several variants of the top-down negotiation algorithm. These algorithms have differed in their maximal and minimal configuration distance and negotiation step limit parameters. The simulation scenario and the simulated service was the same as in the case of the bottom-up negotiation algorithm variant tests. After the first round simulations of many top-down negotiation algorithm variants I have selected four of them that have achieved better performance than the others. In the case of the selected variants the minimal configuration distances were the same as the maximal configuration distances and they are 1, 8, 9 and 10 by value. Similarly to the bottom-up negotiation algorithm variants, the selected variants have not applied negotiation step limit.

In the following figures the top-down negotiation algorithm variants are distinguished using the configuration distance parameter similarly to the bottom-up negotiation algorithm variants.

**Average Teleservice Setup Step and Average Teleservice Adaptation Step Count Performance Metrics for Top-Down Variants**

In Figure 12 and Figure 13 the average teleservice setup step count and average teleservice adaptation step count metrics are shown in the function of the network load for the examined top-down negation algorithm variants.
The curves of the simulation results indicate that the top-down negotiation algorithm variants have rather different performance. Two of them, which configuration distance parameters are 9 and 10, have produced outstanding performance. In the case of the top-down 10 algorithm variant, the average teleservice setup step count metric was always less than two, while in the case of the top-down 9 algorithm variant the same metric was always under 4 steps regardless of the network load. The other two investigated algorithm variants were somewhat worse, but have finished the teleservice setup within 8 and 16 steps providing good enough performance.

**Teleservice Blocking Probability Performance Metric for Top-Down Variants**

Figure 14 shows results of the teleservice blocking probability performance metric in the case of the investigated top-down negotiation algorithm variants in the function of the network load.

The simulation results revealed that the teleservice blocking probability metric is almost identical in the case of the top-down negotiation algorithm variants where the configuration distance parameter is 8, 9 and 10. Among these three variants, the top-down 10 algorithm variant achieves the best performance, but its performance is only slightly better than in the case of the top-down 8 algorithm variant producing the worst performance. According to this performance particular metric, the performance of the top-down 1 algorithm variant is still acceptable, but significantly worse than in the case of the other three variants.
Negotiation Skill Performance Metric for Top-Down Variants

In Figure 15 I have displayed the result of the top-down negotiation algorithm variants in the aspect of the negotiation skill performance metric.

![Figure 15 - Negotiation Skill in the case of the Top-Down Algorithms](image)

The simulation results have shown that the top-down 1 algorithm variant has the most powerful negotiation skill that is always between 80 and 90 percent. The top-down 9 and top-down 10 algorithms have achieved the worst performance among the investigated variants, still their negotiation skills were above 60 percent even in the case of the heaviest network load.

Each of the investigated top-down negotiation algorithm variants has achieved outstanding performance regarding at least one performance metric. In my decision I have appreciated more the algorithm variants that were bests in the case of more performance metrics. Among the bests, the top-down 10 variant delivers the most excellent performance in the view of the average teleservice setup step count, average teleservice adaptation step count and teleservice blocking probability metrics. Nevertheless, I have selected top-down 9 algorithm variant to represent the top-down algorithms in further comparisons, because the performance of the top-down 9 variant was almost identical to the performance of the top-down 10 variant, but its negotiation power was better.

3.5 Comparison of the Teleservice Initiation Techniques

To compare the performance of teleservice initiation techniques, I have simulated four of them: non-adaptive teleservice broker systems, teleservices relying on adaptive transport protocols and adaptive teleservice brokers using the bottom-up and the top-down algorithm. In the case of the adaptive teleservice broker systems, based on the results of the previous investigations, I have used the bottom-up negotiation algorithm with a maximal configuration distance parameter of 2; and the top-down negotiation algorithm whit minimal and maximal configuration distance parameters of 9.

3.5.1 Average Teleservice Setup Step and Average Teleservice Adaptation Step Count Performance Metrics

Figure 16 and Figure 17 show the average teleservice setup step count and average teleservice adaptation step count metrics in the function of the network load. The figures show the results that were obtained in the case of adaptive teleservice broker systems only. Teleservices relying on adaptive transport protocols do not perform resource reservation during the initiation of the teleservice, consequently there are no resource reservation attempts, and thus the average teleservice setup step count metric
is not applicable. In the case of the simulated non-adaptive teleservice broker systems the teleservice broker performs always one reservation attempt only. If the reservation is successful then the service is realized, however, if the reservation attempt is unsuccessful then the broker blocks the teleservice. This way the average teleservice setup step count performance metric is always one independently of the network load.

![Diagram](image1)

**Figure 16 - Average Teleservice Setup Steps**

In the case of the bottom-up algorithm I have got an average teleservice setup step count that is always below 6 steps, while in the case of the investigated top-down algorithm the average teleservice setup step count is always less than 4 steps. For both bottom-up and top-down negotiation algorithms it is true that in the case of light network load there are only a few steps required in average to setup the teleservice. The explanation is that in this network condition a lot of successful reservations happen for the first step and consequently they push down the value of the average teleservice setup step count metric. Similarly, in the case of high network load, the average teleservice setup step count metric begin to fall that I explain with the large number of blocked teleservices requiring only two reservation attempts (i.e. the user’s request and the configuration with the lowest resource demand).

The average teleservice adaptation step count metric has shown that there is a great difference between the bottom-up and top-down negotiation algorithms regarding their adaptation speeds. In the case of an average complexity teleservice, the top-down negotiation algorithm never takes more than 10 negotiation steps in average, while the bottom-up negotiation algorithm requires roughly 22 steps. Considering that the simulated 5 party videoconference service has 5 participants and 4 different media session realization quality for both the audio and video sessions, then it is $16^5$ different valid teleservice configurations. This means that 1048576 configurations should be tested in order to find the optimal configuration. Instead of this, the negotiation algorithms test only a few configurations and deliver the near optimal configuration, which is a quite outstanding performance.

As a conclusion I can state that, in the case of a realistic network using a typical teleservice with an average complexity, the fastest average teleservice setup is achieved by the non-adaptive teleservice broker systems. They always perform only one reservation attempt. Close to the non-adaptive teleservice broker systems, the performance of the bottom-up and top-down negotiation algorithms are also outstanding and quite acceptable in the view of the investigated metrics.
3.5.2 Teleservice Blocking Probability Performance Metric

Figure 18 presents curves for each analyzed teleservice initiation technique showing the value of teleservice blocking probability metric in as a function of network load condition.

![Figure 18 - Teleservice Blocking Probability](image)

Based on the simulation results I can state that the teleservice blocking probability raises as the network load grows, which is natural, since the higher the network load is the less resource is left for teleservices and thus the more teleservice is blocked. I have also shown that the lowest teleservice blocking probability is achieved by the top-down negotiation algorithm regardless of the network load. Non-adaptive teleservice broker systems produce the worst performance. In my explanation due to the lack of mechanisms that avoid teleservice blockings non-adaptive teleservice brokers do not perform configuration adaptations and for this reason the teleservice blocking probability is the highest among all the other simulated approaches. In the case of teleservices using adaptive transport protocols and in the case of the bottom-up negotiation algorithm the teleservice blocking probability is nearly identical. Their performance is significantly better than the performance of the non-adaptive teleservice systems, but they still produce significantly worse teleservice blocking ratio than the top-down algorithm.

The simulation results revealed that in a realistic network using a usual teleservice the top-down negotiation algorithm achieves the lowest blocking probability among the analyzed teleservice initiation techniques and non-adaptive teleservice broker systems has the worst performance. In the case of the other two teleservice initiation techniques the teleservice blocking probability performance is still acceptable.

Despite the outstanding performance of the top-down negotiation algorithm, having extreme high teleservice blocking probability values is not an option for a teleservice. No operator is able to sell a service without reasonable low blocking probability. However, considering a scenario, where there are alternative solutions to reserve resources for a teleservice, the teleservice blocking probability expresses the blocking probability on the cheapest resource reservation method only. If the teleservice setup is unsuccessful here, the service operator has the option to initiate it using a more expensive reservation technique. In this scenario the lowest teleservice blocking probability means the most efficient resource management reducing the cost of the resource reservation.
3.5.3 Teleservice Quality Distribution Performance Metric

To evaluate the teleservice quality distribution metric I have selected four configuration ranges, in which the dominance of the configurations was analyzed. The first range was the group of the blocked teleservices with zero preference. The second range was the group of the low quality teleservices, which preferences have not exceeded 70 percent. The third group was for the high quality teleservices that are more preferred than the low quality teleservices, but their preference was still less than 95 percent. The final quality range was the super quality group where those teleservices were placed that were even more preferred than configurations in the high quality group.

Figure 19, Figure 20 and Figure 21 show the teleservice quality distribution metric using the previously defined quality ranges in the case of teleservices using adaptive transport protocols, in the case of the bottom-up and in the case of the top-down negotiation algorithm. Due to the lack of teleservice configuration adaptation ability, non-adaptive teleservice broker systems are not characterized in the aspect of this metric.
Based on the results of the simulations I have revealed that in the case of teleservices using adaptive transport protocols the dominance of low and high quality teleservices is not as significant as the dominance of the blocked and super quality configuration. This teleservice initiation technique delivers almost equal numbers of low and high quality teleservices, which refers to the lack of the teleservice adaptation method. I have also proven that in the case of the bottom-up negotiation algorithm high quality teleservices are more dominant than the low quality teleservices. In my interpretation this means that the bottom-up negotiation algorithm successfully fulfills the mission of the adaptation producing as much high quality teleservices as it can. In the case of the top-down negotiation algorithm I have shown that using this algorithm low quality teleservices are more dominant than the high quality teleservices.

To see the differences between the teleservice initiation techniques more clearly, in Figure 22, Figure 23, Figure 24 and Figure 25 I present the teleservice quality dominances in separate regions.
Figure 22, the figure of blocked teleservice configurations dominance is the same as it was in the case of the teleservice blocking probability metric describing exactly the same thing. I have presented this figure in this form here to ease the understanding of the teleservice quality distribution metric evaluation. In Figure 23 I have presented the dominance of low quality teleservice configurations. The bottom-up negotiation algorithm produces the smallest number of low quality services that was explained with the efficiency of the negotiation algorithm previously. The highest dominance of low quality teleservices are achieved by the top-down negotiation algorithm, however, considering the small dominance of the blocked teleservices, it is explained that the top-down algorithm adjusts low quality teleservice configurations rather than blocking them. Figure 24 shows the dominance of the high quality teleservice configurations. According to the most efficient negotiation concept, the bottom-up algorithm initiates the largest number of high quality teleservices. The top-down negotiation algorithm delivers less high quality teleservices, while the difference is most significant in the case of teleservices that use adaptive transport protocols, which realize the smallest number of high quality teleservices. The reason of the poor performance in the case of teleservices relying on adaptive transport protocols is the lack of the negotiation coordination that leads to unbalanced and thus less preferred teleservice configurations. Finally, Figure 25 presents the dominance of the super quality teleservice configurations. The dominances in the case of the different investigated teleservice initiation techniques are almost identical. The top-down negotiation algorithm delivers the largest number of super quality teleservice configurations, while teleservices relying on adaptive transport protocols produce the smallest number from the same quality teleservice configurations.
3.5.4 Negotiation Skill Performance Metric

In Figure 26 I present the negotiation skill metric in the case of the bottom-up and top-down negotiation algorithms, in the function of the network load. Since the negotiation skill metric characterizes the negotiation performance of the investigated teleservice negotiation algorithms, this metric is not applicable for non-adaptive teleservice broker systems and for teleservices that use adaptive transport protocols. To evaluate the negotiation skill metric, I have used the brute-force negotiation algorithm that tells the optimal configuration. Each time the teleservice broker performed an adaptation procedure, the brute force negotiation was also performed on the same request in identical network conditions and its result was the basis of the negotiation skill metric.

![Figure 26 - Negotiation Skill](image)

Based on the simulation results I have shown that in the case of the bottom-up negotiation algorithm the negotiation skill performance metric is independent of the network load unlike the top-down negotiation algorithm, of which negotiation skill reduces along the growth of the network load. The bottom-up negotiation algorithm produces excellent negotiation skill that is always 99% regardless of the network load. This underlines my previous explanations that foretold the efficient adaptation capability of the bottom-up negotiation algorithm. In the case of the top-down negotiation algorithm the negotiation skill performance is worse. It has never achieved better negotiation skill as 80%; however it is also true, that the negotiation skill was always above 65%. The relatively low negotiation skill performance expresses that the bottom-up negotiation algorithm delivers more preferred configurations than the top-down negotiation algorithm considering a single teleservice initiation request.

In the view of the simulation results I can state that both the bottom-up and the top-down negotiation algorithms realize teleservice configurations whose preference is suitably close to the preference of the optimal configuration. Even more, the bottom-up negotiation algorithm performs the adaptation almost like an algorithm that provides the optimal configuration.

3.6 Analysis on Alternative Simulation Scenes

In order to assure that the performance characteristics of the investigated teleservice initiation techniques truly describe the teleservice initiation approaches and they are not affected by the lucky or unlucky decision of the simulation scene, I have performed additional simulations modifying the simulated network scenario and the simulated teleservice.
3.6.1 Simulation Using a Alternative Teleservice

First, I have changed the simulated 5 party videoconference teleservice to an 8 party videoconference. In the case of the new service the rules and the timing of the teleservice initiation remained the same as they were in the case of the 5 party videoconference. However this service were a more complex, more demanding teleservice. Considering 8 participants and 16 combinations of the 4 audio and video session qualities, it is over 4 billion valid teleservice configurations. From this view, the 8 party videoconference service is a really complex service.

Similarly, as in the case of the previous simulations, I have compared four teleservice initiation techniques to each other: non-adaptive teleservices; teleservices relying on adaptive transport protocols; and the proposed adaptive teleservice broker system using the bottom-up and top-down algorithms. Before the comparison, I have determined the configuration distance and negotiation step limit parameters of the negotiation algorithms analyzing different variants. As a result, I have selected the bottom-up negotiation algorithm variant that applied 2 as its maximal configuration distance parameter and the number of negotiation was limited to 100 steps. In the case of the top-down negotiation algorithm, the minimal and maximal configuration distance parameters were 15 and this algorithm applied no negotiation step limit.

Since the 8 party videoconference initially has higher network resource demand than the 5 party videoconference teleservice, it is evident that in the case of lighter network load there are higher teleservice blockings and the investigated performance metrics reflect its effect. This is the reason, why I have performed the simulations on a 57% to 99% network load range, while previously I have investigated the 85% to 99% network load domain.

Figure 27 and Figure 28 show the result of the simulations investigating the average teleservice setup step count and the average teleservice adaptation step count metrics. The performance of the negotiation algorithms are slightly worse than in the case of the 5 party videoconference teleservice, however, considering that in this case, there are around 4000 times more potential teleservice configurations, then the results are quite outstanding. Here again, the top-down negotiation algorithm achieves extremely good performance and the performance of the bottom-up algorithm is also acceptable.

The results of the teleservice blocking probability metric evaluation can be seen in Figure 29. These results are very similar to the case when I have examined the teleservice initiation approaches using the 5 party videoconference teleservice. The lowest teleservice blocking probability is achieved by the top-down negotiation
algorithm. The bottom-up negotiation algorithm and the teleservice initiation approach that rely on adaptive transport protocols performs roughly the same, while finally, the non-adaptive teleservice broker system produces the weakest performance.

![Figure 29 - Teleservice Blocking Probability](TEN-155 Network Scenario, 8 Party Videoconference)

Figure 30, Figure 31, Figure 32 and Figure 33 present the dominance of the realized teleservice configurations in the different quality ranges. The quality ranges were the same as in the previous simulations: blocked teleservices, low, high and super quality teleservices. The simulation results have shown that there is no change in the tendencies in the case of the negotiation algorithms. The bottom-up negotiation algorithm makes all effort to realize as high quality teleservice as it can, while the top down negotiation algorithm delivers the lowest quality teleservices, the lowest number of blocked teleservices and the largest amount of super quality teleservice configurations. However, in the case of the teleservice initiation approach that relies on adaptive transport protocols, there is a performance reduction compared to the previous results. While in the case of an average complexity teleservice this teleservice initiation technique balances the number of low quality and high quality teleservices, then in this case, using a more complex teleservice, the low quality teleservices become more dominant.
Finally, I have examined the negotiation skill performance metric in the case of the two teleservice negotiation algorithms. Figure 34 show the negotiation skill metric in the function of the network load. Based on these results I can state that changing the teleservice to a more complex one does not influence the bottom-up negotiation algorithm, which negotiation power is still excellent, delivering teleservice configurations that are 99% close to the optimal configuration. However, complex teleservices reduce the negotiation power of the top-down algorithm in the case of lighter network load conditions. Still, the performance of the top-down negotiation algorithm is acceptable in the aspect of the negotiation skill metric.
As a conclusion I can state that introducing a more complex teleservice does not change the ranking among the investigated techniques. Although, due to the higher network resource demands of the more complex teleservices, there are already teleservice blockings in lighter network load conditions, but the shapes of the performance metrics do not change. The exception is the teleservice initiation approach that relies on adaptive transport protocols, since in this case the more complex the teleservice is, the more dominant are the low quality teleservice configurations. In addition, due to the more complex teleservices, the negotiation skill of the top-down negotiation algorithm is reduced.

3.6.2 Simulation on an Alternative Network Scenario

In the previous simulations I have investigated a network scenario that is effective as a continent wide network. Now, the alternative network topology, presented in Figure 35, copies the structure of a country wide network simulating backbones and access routers in a tree topology. This simulated network consists of a 7 node backbone, which backbone routers are connected to each other using 155 Mbps links. Moreover, each backbone router residing on the border of the network connects 4 access routers on 55 Mbps capacity links, plus the access routers connect 500 terminals each. Thus, in total, there are 7 backbone routers, 20 access routers and 10000 terminals.

Using this alternative network topology and the 5 party videoconference teleservice, I have evaluated the performance of the four investigated teleservice initiation techniques. So as with the case where I have changed the teleservice, I have
performed first round simulation in order to select those bottom-up and top-down negotiation algorithm variants that represented the adaptive teleservice broker system in the comparison. As a result of these simulations I have chosen the bottom-up negotiation algorithm whose maximal configuration distance parameter is 2; and the top-down negotiation algorithm with a minimal and maximal configuration of 9. In the case of both variants, there was no negotiation step limit specified.

Due to the modification of the simulated network topology, the amount and the distribution of the network resources also has changed. According to the new situation I have performed analysis on the 77% to 99% network load domain. This was the network load range where the blocked teleservices have shown their impact on the investigated performance metrics.

Figure 36 and Figure 37 presents the simulation results that measure the negotiation speed of the bottom-up and top-down algorithms. In spite of that the range of the analysis is different compared to the TEN-155 simulation scene, the shape of the curves are almost identical. The most significant difference is that in the case of heavy network load conditions both the bottom-up and top-down negotiation algorithm take more adaptation steps than in the case of the TEN-155 simulation scenario. However, the conclusions remain the same as before: both teleservice negotiation algorithms have outstanding performance according to these metrics; and the fastest algorithm is the top-down negotiation algorithm.

The simulation results in the case of the teleservice blocking probability metric are presented in Figure 38. Based on these results I can state that it is still the top-down negotiation algorithm that bears with the lowest teleservice blocking ratio; the performance of the bottom-up and the teleservice initiation approach that relies on adaptive transport protocols are identical; and finally, the weakest approach is the non-adaptive teleservice broker systems in the aspect of this performance metric.
Figure 38 - Teleservice Blocking Probability
(Country-wide Network Scenario, 5 Party Videoconference)

Figure 39, Figure 40, Figure 41, and Figure 42 show the dominance of the delivered teleservice configurations in different quality ranges in the function of the network load. Comparing the tendency of the results to the ones obtained in the case of the TEN-155 network scenario, I have shown that there is no significant change, except the case of the teleservice initiation technique that uses adaptive transport protocols. In this particular case, the dominance of the low and high quality teleservices that were balanced in the case of the TEN-155 network scenario, becomes unbalanced as the low quality teleservice configurations grows to be more dominant.

Figure 39 - Configuration Dominance of the Blocked Teleservices
(Country-wide Network Scenario, 5 Party Videoconference)

Figure 40 - Configuration Dominance of Low Quality Teleservices
(Country-wide Network Scenario, 5 Party Videoconference)

Figure 41 - Configuration Dominance of High Quality Teleservices
(Country-wide Network Scenario, 5 Party Videoconference)

Figure 42 - Configuration Dominance of Super Quality Teleservices
(Country-wide Network Scenario, 5 Party Videoconference)

At last, I have evaluated the two teleservice negotiation algorithms in the aspect of the negotiation skill performance metric. Figure 43 show the results of the analysis in the
function of the network load. The results suggest that changing the network scenario does not change the negotiation power of the examined algorithms. The bottom-up algorithm delivers teleservice configurations that are 99% close to the optimal one, while the performance of the top-down algorithm is above 70% almost everywhere, which is still acceptable.

![Figure 43 - Negotiation Skill](image)

(Country-wide Network Scenario, 5 Party Videoconference)

Based on the results of the simulations I have proven that changing the network scenario in a rational way does not change significantly the characteristics of the investigated teleservice initiation techniques. The only exception is the teleservice initiation approach that relies on adaptive transport protocols. This technique produces more balanced low and high quality teleservice configuration dominance in the case of networks, having more alternative routes, such as TEN-155, than in the case of networks in a tree structure.

3.7 The Required Computing Capacity of the Negotiation Algorithms

The performance analysis of teleservice initiation techniques has proven that the adaptive teleservice broker systems produce better performance than the other teleservice initiation approaches according to the majority of the performance metrics. However, network and service providers might profit from the benefits of the proposed adaptive teleservice broker system only if this approach bears with an affordable complexity. For this reason I have performed examinations, where I have investigated the required computing capacity of teleservice initiation. Since the most complex part of the adaptive teleservice broker system is the teleservice negotiation procedure, I have measured the processing requirement of the bottom-up and top-down negotiation algorithms. The complexity of the other procedures in the teleservice broker is negligible.

During the negotiation procedure both algorithms construct various configuration sets, the extended and the downgraded configuration sets, from which they choose certain configurations to tests against the network resources. Since the construction of a teleservice configuration is based on a lot of computations, I have described the complexity of the algorithm by the number of teleservice configurations that were assembled throughout the whole negotiation.

Figure 44 and Figure 45 display the average complexity, which is the average number of configurations that were constructed during the negotiations. The results are
presented in the function of the network load, both for the 5 party and the 8 party videoconference services using the TEN-155 network scenario.

![Figure 44 - Average Complexity of the Algorithms in the case of the 5 Party Videoconference Service](image)

![Figure 45 - Average Complexity of the Algorithms in the case of the 8 Party Videoconference Service](image)

The results of the simulations have proven that the teleservice negotiation algorithms do not demand extreme computing capacity, the complexity is acceptable and thus the adaptive teleservice broker system is feasible.

Moreover, the adaptive teleservice broker is designed to be a standalone application and therefore the network or service operators have the ability to place more than one teleservice brokers into their networks. This way the teleservice initiation requests can be balanced among all the adaptive teleservice brokers, reducing the overall complexity that the teleservice initiations demand.

### 3.8 Overall Conclusions

It is hard to judge which negotiation algorithm is better, since none of the investigated teleservice initiation techniques is the ultimate winner of the performance analysis. This way, the judgment depends on the aspect of the examiner. Over all, I can state that all adaptive teleservice initiation techniques realize better teleservice blocking probability than it is in the case of non-adaptive teleservice broker systems. Since the revenue of the network seriously depends on teleservice blocking probability, I recommend using adaptive teleservice techniques, especially the adaptive teleservice broker with the top-down negotiation algorithm.

In the case of adaptive teleservice broker systems I have proposed two negotiation algorithms, of which performance characteristics are rather different. The bottom-up negotiation algorithm delivers teleservices that are really close to the optimal teleservice configuration, but as a tradeoff in the aspect of all the other performance metrics it performs behind the top-down negotiation algorithm. In contrary, the top-down negotiation algorithm almost unbeatable according to nearly all performance metrics, but considering a given teleservice initiation request, the adapted teleservice is less preferred than it is in the case of the bottom-up negotiation algorithm. In line with the advantages and disadvantages of the proposed negotiation algorithms I recommend using the bottom-up negotiation algorithm in networks, where the network is optimized to the number of high and super quality teleservices; and I suggest to apply the top-down algorithm in those networks where the low blocking probability and the fast teleservice setup is most preferred against the expected quality of the requested teleservices.
3.9 The Future of Adaptive Teleservice Broker Systems

In future networks, where end-to-end QoS provisioning is not a problem anymore, the usage of adaptive applications, transport protocols and teleservice broker systems might seem to be unnecessary. However, even in this case, the adaptive teleservice broker systems enhance the quality of the teleservices as well: In the QoS enabled networks when a teleservice configuration does not go through the resource reservation due to a temporary resource shortage then the user has to modify the configuration and resubmit the initiation request itself. Transferring this negotiation procedure inside to the network decreases the teleservice setup time as it requires no user interaction. On the other hand, despite the QoS guarantees that the network operator offers to the customers, there might also be physical or other (e.g.: business or political) reasons when even the QoS enabled networks are unable to provide the dedicated resources (just think of the mobile networks). In these situations reconfiguring the teleservice via an adaptive teleservice broker is also advantageous.
4 Analysis of Routers Supporting Resource Reservation

Bandwidth is in the process of becoming a cheap commodity today. Both network operators and users have more and more of it at their disposal, and as available bandwidth increases, so does the demand for even more bandwidth. However, in some cases it is either impossible or infeasible to satisfy this hunger by deploying more physical bandwidth (e.g. mobile, ad-hoc or wireless networks). Traffic engineering may, in some cases, alleviate problems for a while, but operators are still often forced to rely on IntServ-based [1][3] or DiffServ-based [4] QoS provisioning.

There are more and more QoS-aware IP router products available today. These routers have rather different capabilities in terms of signaling and forwarding performance. Recently, there has also been a large palette of IP resource reservation protocols: the ancient ST-II [29], the most famous RSVP [30]; some less wide known protocols: DRP [31], YESSIR [32]; lightweight approaches: Boomerang [C3], and Load Control [33] and others. However, regardless of the wide range of router products and versatility of reservation protocols the end-to-end QoS provisioning is still not supported in the Internet. It is a common opinion that one of the main reasons for the lack of end-to-end QoS provisioning is that network operators are not aware of the performance characteristics of the resource reservation capable routers and the impact of resource reservation signaling on the overall QoS performance of the network. There already exist some performance studies, such as [C3], [42] and [43] that investigate the scalability of certain router products, but operators still miss the possibility to compare the performance of several reservation capable routers in order to design QoS enabled networks.

In order to give a tool to investigate the performance of reservation capable routers, I have set up a performance analysis framework, in which the reservation signal handling and traffic forwarding performance characteristics of a router can be examined at the same time. The proposed performance analysis framework is designed to work for arbitrary router products. The universality is achieved by using a resource reservation capable router model that is built up from building blocks that all resource reservation capable routers implement. Originated from this router model I have determined those parameters that are able to affect the performance of the routers. Based on these parameters I have established the profiles of the performance characteristic equations. These performance characteristic equation profiles are parametric equations describing the performance metrics containing the parameters that affect the load on the router. The exact performance characteristic equations can be determined by benchmarking measurements, for which I have given a methodology and also presented two examples. Later, the performance characteristic equations are suitable to be used in network simulations to analyze the resource reservation handling performance of the whole network; however this analysis was out of the scope of my dissertation.

During the investigation of the resource reservation protocols I have studied single processor resource reservation capable routers running single threaded resource reservation protocol implementations only. For this reason the proposed performance analysis framework is suitable for single processor routers using single threaded resource reservation protocol implementations. Despite that this is a significant limitation to the performance analysis framework, my experiences show that most
dedicated routers also use a single processor to perform resource reservations and usually this processor is also affected in the traffic forwarding. In addition, among the available resource reservation protocols there are hardly any multi-threaded implementations. Thus the proposed performance analysis framework is suitable for a wide range of resource reservation capable routers.

4.1 Resource Reservation Capable Router Model

In order to build up a resource reservation capable router model I have examined various router architectures and open source code resource reservation protocol implementations. Despite the huge variety of the examined resource reservation protocols, I have ascertained that certain features of a protocol are common with features in other protocols. Some of these resource reservation protocol features refer to the basic behavior of the protocol while other features determine the exact method or the parameterization of the resource reservation only. Analyzing the functions of the key features, I have described what kinds of building blocks are necessary to implement a certain feature of the protocol. Later, to achieve a universal resource reservation capable router model, that is the key to a successful performance analysis framework, I have synthesized all the identified building blocks into a single model. This assures that the router model is generic enough to be suitable for each resource reservation protocol, since this model contains all the building blocks that are necessary to implement all the features of an arbitrary protocol. Naturally, when a resource reservation protocol does not implement a certain feature that is in connection with some of the building blocks then the modeler simply ignores these building blocks.

4.2 Resource Reservation Protocol Features

I have investigated several resource reservation protocols to identify the basic features that determine the main implementation concepts. Of course, each resource reservation protocol has unique implementation, but still, there are certain activities in relation to some features of the protocol that perform similar jobs and therefore the implementation is similar as well. During the investigation of the various resource reservation protocols I have collected only the key features of the resource reservation protocols. These features seriously affect the implementations:

- **Hard-State Protocols**: Basically, this kind of resource reservation capable routers establishes permanent states for a resource reservation session. In the case of successful reservation on the whole path of the session, the allocated states remain active forever, unless a signaling message requests the explicit tear of the reservation session. Consequently, the resource reservation sessions can be removed only explicitly. In this form, resource reservations performed by hard-state resource reservation protocols are very similar to the virtual circuits that ATM switches build.

- **Soft-State Protocols**: In the case of the soft-state protocols the main objective was to provide robustness to the resource reservation protocols. For this reason, even in the case of successful resource reservation the allocation remains active only for a certain time. After this interval, the allocation is canceled unless a special signaling message, called refresh message, arrives and refresh the states of a particular reservation session. Consequently, in the case of the soft-state protocols, the reservation session is terminated if no
refresh request arrives during the refresh period or a signaling message requests the explicit tear of the reservation session. The soft-state name refers to the temporary position of the reservation sessions.

- **Stateless Protocols**: Stateless resource reservation protocols do not distinguish reservation sessions for the allocated resources, reducing the complexity of the protocol. Similarly to the soft-state protocols, in this case, the resource allocations are valid only within a certain interval and the reservation request should be repeated once in each new interval. The lack of the repeated reservation request or a signaling message requiring the explicit reservation tear down signs the end of the resource allocation. The stateless name reflects that the router have no stored states for reservation sessions.

- **Protocols with Explicit Call Admission Control** [48]: In the case of such resource reservation protocols, the routers keep track of all the resource reservation requests and from this information they compute the amount of free and allocated resources. A new resource reservation request is successful only if the router has enough free resources.

- **Protocols with Measurement Based Call Admission Control**: Unlike protocols with explicit call admission control, in the case of measurement based call admission control, the router determines the admission of a new resource reservation request based on measurements performed at the moment of the decision. This call admission control algorithm requires no information about the resource reservation sessions.

- **Protocols with Multiparty Reservation Session Support**: These kinds of resource reservation protocols support multiple senders and multiple receivers in a single reservation session. The resource reservation protocol often supports several resource reservation schemes that control the resource sharing among the routers that participate in the whole multiparty reservation session.

- **Receiver Oriented Protocols**: In the case of resource reservation protocols supporting multiparty resource reservation sessions it is a focal question that who maintains the list of the participants in the session. If it is on the sender side of the traffic flow then the senders should take care of all the joining and leaving requests that could be a heavy duty in the case of a large multiparty reservation session. The other option is that the receivers of the traffic flow individually take care of their own session joining and leaving activities lightening the load on the traffic sources. In the case of receiver oriented protocols it is the data traffic receiver who initiates the resource reservation setup and tear down requests. The resource dedication should always take place on those routers that are involved in the traffic forwarding from the senders towards to receivers, thus receivers should know the path of the data traffic before they can send any resource reservation requests in the opposite direction. Consequently, senders should mark the path of the traffic flow before the resource reservation. This requires additional signaling messages, the path setup and tear down message pair that highlights the path of the reservation.

- **Sender Oriented Protocols**: In the case of sender oriented protocols the data traffic senders initiate the resource reservation requests and this way they follow the path of the data traffic without any extra path marking. There are
also protocols where the receivers are capable of initiating a resource reservation request instead of the senders. In these protocols the signaling messages are transferred to the sender side first and then the resource reservation setup or tear down request travels back from that point. In this case, the resource reservation is initiated by the receiver, but since the signaling messages travel from the sender to the receiver, I consider these protocols as sender oriented protocols.

**Instant Response Protocols:** Some of the recent lightweight protocols do not require the routers to initiate signaling messages (such as refresh, error or information messages) to other routers or to the signaling endpoints. Instead, the signaling messages are always generated by signaling endpoints or signaling proxies and the information that a router would like to tell to other routers, such as the status of the reservation, are piggybacked to the signaling messages. Of course, these routers are able to send messages in one direction only.

Based on these basic features of the resource reservation protocols I have classified each investigated resource reservation protocol:

**ST-II [29]:** The ST-II (Internet Stream Protocol Second Version) protocol is the only widely known hard-state resource reservation protocol. In spite of it is a standardized resource reservation protocol, it had failed to dominate the Internet. The reasons are the extreme complexity of the protocol and the fact that Internet routers are not designed for 100% stability, thus they are not robust enough to maintain hard states that is one of the bases of the ST-II protocol. The ST-II is a sender oriented protocol and supports multiparty resource reservation sessions without the assistance of any multicast transport protocol. It is designed to build up a multicast tree among the participants of a multiparty resource reservation session, and then both the signaling messages and data traffic are forwarded along this multicast tree. The ST-II protocol applies explicit call admission control on each new reservation request.

**RSVP [30]:** RSVP (Resource Reservation Protocol) is the most famous resource reservation protocol today. It was standardized in 1997. Unlike ST-II, RSVP is built up on soft-states that ensure the robustness over the unpredictable Internet. It supports multiparty resource reservation sessions, which are based on different reservation schemes controlling the resource distribution among the participants. Due to the design concept that in the case of a huge multiparty reservation session the fluctuation of the destinations is more significant than the rise and fall of the data sources, RSVP is a receiver oriented protocol. Regarding the method of the call admission control, RSVP relies on the routers’ traffic control routine that performs either explicit or measurement based checking on the new resource reservation requests.

**DRP [31]:** The Dynamic Sender-Initiated Reservation Protocol is a sender-oriented soft-state resource reservation protocol. In the case of this resource reservation protocol, the designers have voted for the sender orientation in order to reduce the complexity, the number of signaling messages and the number of states in the router. The protocol is capable of reserving multiparty resource reservation sessions and similarly to the RSVP, it uses resource reservation schemes to distribute resources among the participants. This resource reservation protocol requires explicit call admission. Although in
most of the cases the signaling messages are forwarded immediately by the router, there are some special situations when the router still initiates signaling messages.

**Boomerang [C3]:** The Boomerang protocol is a light-weight soft-state resource reservation protocol. The light-weight approach means that the primary goal of the authors was to construct a really fast resource reservation protocol, even if it has limited functionality. In order to decrease the complexity, the Boomerang resource reservation protocol is sender oriented, however it also supports a mechanism that allows the receivers to initiate resource reservation requests transferring their signaling messages to the sender side first. Moreover, Boomerang is an instant response protocol and signals to other routers by piggybacking the information to signaling messages. Boomerang generates no signaling messages, consequently it is also unable to send information backward. For this reason, the Boomerang protocol is able to reserve special multiparty sessions only, in which there is only one sender considering the whole group. The Boomerang protocol applies explicit call admission control.

**YESSIR [32]:** Features of the YESSIR (YEt another Sender Session Internet Reservations) protocol are identical to the DRP protocol. However, there is a slight difference. Namely, this protocol assumes that the transport protocol of the reserved sessions is the RTP [11] protocol. This way, there are no explicit resource reservation setup and teardown signaling messages, but routers examine the control messages of the RTP protocol and perform the reservation actions accordingly.

**Load Control [33]:** The Load Control protocol is also a light-weight approach for resource reservation. In this case, the low complexity is achieved by removing the resource reservation states from the protocol. Thus the Load Control protocol is a stateless protocol. Having no path or session information, the protocol is necessarily sender oriented. Since it does not require complex signaling messages, it is enough to send router information in forward direction, and therefore it is an instant response protocol. Currently, this protocol applies explicit call admission control, however in the future there will be also an alternative version that relies on measurement based admission control.

Table 4 summarizes the resource reservation protocols and their key features.
Table 4 - The Basic Features of Resource Reservation Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>States</th>
<th>CAC</th>
<th>Multiparty</th>
<th>Orientation</th>
<th>Instant response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-II</td>
<td>hard</td>
<td>explicit</td>
<td>yes</td>
<td>sender</td>
<td>no</td>
</tr>
<tr>
<td>RSVP</td>
<td>soft</td>
<td>depends on the</td>
<td>yes</td>
<td>receiver</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRP</td>
<td>soft</td>
<td>explicit</td>
<td>yes</td>
<td>sender</td>
<td>no</td>
</tr>
<tr>
<td>Boomerang</td>
<td>soft</td>
<td>explicit</td>
<td>limited</td>
<td>sender+</td>
<td>yes</td>
</tr>
<tr>
<td>YESSIR</td>
<td>soft</td>
<td>explicit</td>
<td>yes</td>
<td>sender</td>
<td>no</td>
</tr>
<tr>
<td>Load</td>
<td>stateless</td>
<td>depends on the</td>
<td>no</td>
<td>sender</td>
<td>yes</td>
</tr>
<tr>
<td>Control</td>
<td></td>
<td>implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3 The Building Blocks of Resource Reservation Protocols

Once I had the basic features of the resource reservation protocols, I have analyzed them to determine what kinds of building blocks are necessary to implement them. Using these building blocks one can implement a resource reservation protocol simply by choosing a conforming set of the required features and assembling the building blocks that are in connection with the features. Similar project already exist for standard router devices without resource reservation support [49].

I have gone through the key features of the resource reservation protocols and after analyzing them I have determined, what kinds of building blocks are necessary to implement them:

- **Hard-State Protocols**: Since hard-state protocols distinguish each flow that has resource dedication in the router, the router must provide a mechanism to differentiate the resource reservation sessions. This is called session lookup procedure and thus every hard-state protocols must implement a session lookup block.

- **Soft-State Protocols**: Soft-state protocols just as hard-state ones differentiate the resource reservation sessions and therefore they also implement the session lookup block. Moreover, in the case of soft-state protocols the resource reservation sessions have to be maintained by refresh signaling messages and thus these protocols also utilize a session maintenance block.

- **Protocols with Explicit Call Admission Control**: In the case of such protocols the call admission control is based on explicit calculations that consider other resource reservation sessions and tell whether the new session gets enough free resources or not. This is the task of the call admission control block [48]. When the call is accepted by the router then a certain amount of resources must be dedicated for it. This is performed by the resource dedication block that parameterizes the traffic control for the session. Routers usually have more than one outgoing links and therefore the resource reservation capable router must know on which link resource dedication happens. For such reason these protocols implement a route query block, which provides the outgoing interfaces that are affected in the actual resource reservation session.
Protocols with Measurement Based Call Admission Control: Similar to the task of the explicit call admission control, measurement based call admission control protocols should implement a call admission control block, a resource dedication block and a route query block as well. However, in this case the call admission control block performs usually measurements and explicit calculations.

Protocols with Multiparty Reservation Session Support: The proper handling of the multiparty reservation sessions requires routers to initiate signaling messages informing other routers involved in the actual reservation session. This task has a need of a signaling message initiation block.

Non Instant Response Protocols: Resource reservation protocols that are not instant response protocols have to initiate signaling messages. This task is also performed by the signaling message initiation block.

Receiver Oriented Protocols: Receiver oriented resource reservation protocols have to maintain the path of the data traffic beside the path of the reservation. This requires extra session and state information that multiply the task of the session lookup, session maintenance and signaling message initiation blocks.

Naturally, new resource reservation proposals might bring up new ideas, features and consequently new building blocks might appear as well. In my dissertation I have identified and described only building blocks, from which existing resource reservation protocols are built up.

Since some of the building blocks are always in companion with other building blocks, I have made groups containing the blocks. Combining the building blocks with the path of the data traffic and signaling messages within the resource reservation capable router, I have differentiated the following main building blocks:

4.3.1 The Traffic Forwarder Block

The traffic forwarder block is responsible to route and forward the incoming data packets towards the destination of the data flow. Usually advanced router architectures allocate multiple processors for routing and forwarding data packets [41], but even in this case the routing table computation and the propagation of the routing information are dedicated to one central processor unit. Naturally, in router architectures where there is only one processor (e.g.: PC based Linux routers or low-end routers with bus architecture) all the forwarding and routing related work is performed by the single central processor. In addition, in the case of IntServ routers, the priority data packets having resource allocations in the router should be associated with the corresponding reservation sessions sessions first. The mapping between the prioritized data packets and the reservation sessions is usually not the task of the traffic forwarding units. Based on these facts, I can state that in the case of most recent router products, the traffic forwarding and the signaling message processing have influence on each other. Since routers are generally designed to data forwarding, the traffic forwarder block is prioritized on some architecture (e.g.: Linux).

I have highlighted two blocks inside the traffic forwarder block. First, the traffic identification block that investigates data packets and if a packet claims for priority handling, then it tries to associate the corresponding session to it [44]. Second, the...
routing and forwarding block that performs any other necessary actions to place the packet into the buffer of the appropriate outgoing interface.

### 4.3.2 The Signaling Message Processing Block

Beside the traffic forwarder building block there is the signaling message processing block that handles the signaling messages. According to my assumptions, this building block has three main functions: (i) it interprets the incoming signaling messages; (ii) manages the resources; and (iii) initiates signaling messages. In line with these functions I have further derived the signaling message processing block to smaller blocks:

1. **Session Lookup:** The signaling message interpretation begins with the identification of the signaling message. In the case of hard-state and soft-state protocols the signaling message identification also involves the identification of the reservation session that the signaling message refers to. The session assignment is performed by the session lookup block that can be found in every hard-state and soft-state resource reservation protocols, but not in stateless protocols, since these protocols does not differentiate reservation sessions [45]. Moreover, in the case of receiver oriented protocols it is not enough to look up the corresponding reservation sessions only, but the information about the reservation paths also should be acquired. The identified signaling messages accompanied by the corresponding reservation sessions and other reservation state information are forwarded to the resource management block. The only exceptions are the refresh signaling messages that are forwarded to the session maintenance block.

2. **Resource Management:** The resource management block performs the required actions on resource allocations that the signaling messages specify. Usually the resources of the router are in relation with the buffers of the outgoing interfaces. This way, if the signaling message requires some actions manipulating the resources that are dedicated to the corresponding session then these resources should be identified first. The route query block tells all the outgoing interfaces of the actual reservation session in a router. In spite the place of the resource allocation always should be known for the router, not all the resource reservation protocols require the implementation of the router query block. There are protocols, such as the Boomerang protocol, in which the signaling message processing block is hooked to the outgoing interfaces and therefore it gets all the signaling messages after the routing when the place of the reservation is already known.

When a request is about to allocate new resources for a resource reservation session then the router tests the resource demand of the request against the free resources. This test is performed by the call admission control block that can be explicit or measurement based. The explicit call admission control algorithm considers the demands of other resource reservation sessions as they describe themselves and calculates the amount of the free resources from this information. In contrary, the measurement based call admission control algorithm does not need information about other resource reservation sessions, because this algorithm performs passive measurements to estimate the free resource capacities. In both cases the new resource reservation session can be
admitted only if the demand of the new session is smaller than the calculated or estimated amount of free resources.

If the actual signaling message carries a request to alter the amount of dedicated resources or to allocate a new reservation session then the router performs these manipulations through the resource dedication block. In the case of IntServ based protocols the task of the resource dedication block is to modify the traffic control parameters of the resource reservation sessions. These traffic control parameters specify the prioritization of the traffic flow that refers to the session and also limit and occasionally shape this traffic. In the case of DiffServ based protocols the task of the resource dedication block is much simpler, since the traffic is already marked with the corresponding prioritization sign and this does not affect the traffic control.

3. **Signaling Message Initiation**: Normally, the processed signaling message is required to be forwarded to other routers as well. The exceptions are only those signaling messages that address the actual router or when the protocol describes that the router have to drain the signaling message. In the case of instant response protocols, the signaling message forwarding does not require extra building block inside the signaling message processing block, since the signaling messages are simply forwarder toward their destinations. Even in this case, the routers have the chance to broadcast error and other information to other routers by extending or altering the signaling messages before the forwarding. However, non instant response protocols require sending signaling messages to other routers even if no signaling message is processed. For this reason the router has to construct and send the appropriate signaling message to the destinations. This is the task of the signaling message initiation block.

Usually recent routers do not dedicate separate processor for the signaling message processing task, but instead it runs on the central processor of the router sharing the processing power resource with other competing tasks. There are certain advantages to do this. Since routers usually have a small operating system, the resource reservation protocol implementation becomes simpler, moreover the vendor is able to upgrade or it easily. However, the limitation of this solution is that the performance of the signaling message processing task is affected by other tasks.

### 4.3.3 The Session Maintenance Block

At last, all soft-state resource reservation protocol implementations have a key component, the session maintenance block that is responsible to maintain the resource reservation sessions already allocated in the router. Soft-state routers, due to the robustness, are designed in a way that the reservation sessions have to be refreshed periodically. If a reservation session have not refreshed for a longer time interval, then the router supposes that the reservation session is cancelled without explicit notification and therefore it tears down the session. Consequently, to maintain the reservation refreshes, the session maintenance block watches the expiration of the soft-state sessions and periodically initiates session refresh messages to signal the reservation session refresh to other routers. However, in the case of instant response protocols, there is no possibility to initiate refresh signaling messages by the router itself. In this case the signaling end-points provide the necessary refresh signaling messages and the routers only forward them along the signaling path.
The session maintenance block usually runs on the same processor as the signaling message processing block causing competition for the processing power and other resources.

4.4 The Router Model

Using the identified building blocks I have synthesized a resource reservation capable router model. The model is presented in Figure 46. This model contains all the building blocks that I have derived from the basic features of the resource reservation capable protocols and thus it is able to describe a router that bears an arbitrary but conforming set of such features. However, as I mentioned earlier, I have considered single processor only, and therefore the router model suitable for single processor routers running single threaded resource reservation protocols.

![Figure 46 - Resource Reservation Capable Router Model](image)

In the router model, among the building blocks there are some that perform their job in parallel with other tasks competing for various resources of the router, such as the processing power, memory buffers, data transfer on system buses. To control the access to these common resources, the routers apply scheduling algorithms that manage the internal resource dedications to different tasks. Using the proposed router model I have kept in mind that this resource scheduling has great impact on the signaling message handling and traffic forwarding performance of the router.

4.4.1 The Validation of the Router Model

Since the router model is built up on my assumptions about the existence of the building blocks determining the key features of the resource reservation protocols, I have investigated the source code of some resource reservation protocols. As a result of the investigation, in each resource reservation protocol, for which I have the source code (e.g. RSVP), I have identified all the building blocks that the features of the protocol indicated. Thus, I have validated my theory about the existence of the building blocks. Besides, the same examination validated that the construction of the router model and the connection between the building blocks are also close to the reality.

4.4.2 Similar Router Models

Nowadays there are many existing IP router model publications [34][35][36][37], however none of them touches the resource reservation capability of routers. For this reason, I cannot compare my router model to other router models. On the other hand, I
have found similar models describing ATM switches [38][39]. ATM switches differ from IP routers conceptually, but in the aspect of the resource reservation procedure they share some common features as both of them take similar actions to set up a reservation session. Figure 47 shows a model for general call processing in ATM switches [40].

Figure 47 - Call Processing Model of an ATM Signaling Point

Comparing this model to my proposal is might look like comparing apples and pears, but concerning the similarities between the resource reservation features there is a sense to do so. The similarity is that both models are built up from building blocks which are driven by signaling messages in a specified sequence. The difference between the models is that in the case of the ATM call processing model each building block has a unique processor, while my model describes competition among the blocks assuming a scheduler that controls the processor allocation among the tasks. The reason for the different modeling approach is based on the physical architectures of the network devices. While the first approach assumes that there are special hardware elements for each building block running in parallel, then the second model supposes that the building blocks are software elements running on a single processor.

4.5 Performance Metrics

The focal question of most of the benchmarking measurements is: which measurement methods are the most suitable for benchmarking? The first aspect to be considered is what can one measure on a router product. The question arises because router vendors usually do not allow users to look inside their product and perform measurement on some parts of the router device. Although, in the case of recent router products there are some performance statistics [46] that are available via management protocols, such as SNMP [47], unfortunately, most of this information covers the forwarding performance of the router and says nothing about the resource reservation handling performance that is one of the main target of the analysis. For this reason, I specify performance metrics that are able to characterize the joint resource reservation handling and traffic forwarding performance of the routers from outside of the device treating them as a black box during the measurements.
The most widely used method to characterize a router that handles signaling messages, is to measure the processing time of the signaling messages. The processing time tells about the complexity of the signaling message processing and also able to express the influence of other factors that have impact on the signaling message processing task. The measurement is simple, even in the case when the router product does not support information about it. Assuming that the signaling message does not have to wait before and after the processing, the signaling message processing time equals to the time that a signaling message spends inside the tested device. Thus, the signaling message processing time metric can be measured easily from outside of the tested router by recording the entering and leaving time of a particular signaling message and calculating the difference between these two timestamps.

Since the resource reservation capable routers, especially the ones that have only one processor, perform signaling message processing and data forwarding simultaneously, these two tasks have impact on each other. For this reason, a second metric characterizing the data forwarding performance, should be involved in the performance analysis as well. In order to describe the data forwarding performance, I have chosen the data traffic forwarding time metric, which is the delay that a data packet suffers while it is forwarded by the router.

In summary, I have selected the following two traditional performance metrics for the proposed performance analysis framework that are able to characterize resource reservation routers efficiently and also can be simply and accurately measured.

**4.5.1 Average Signaling Message Processing Time**

The signaling message processing time is the time that a signaling message spends inside the router while the router works on it. Measuring the processing time of a signaling message passing through an unloaded router explores the complexity of the implementation of signaling message handling. On the other hand, taking measurements on loaded routers has led me to study how other tasks influence the procedure of the signaling message processing. The average signaling message processing time \( t_s \) can be calculated as follows:

\[
 t_s = \frac{\sum_{i=1}^{n} s_{i}^{\text{out}} - s_{i}^{\text{in}}}{n},
\]

where \( n \) is the number of the measurements, \( s_{i}^{\text{in}} \) expresses the time instant when the \( i \)th signaling message appeared on one of the incoming ports and \( s_{i}^{\text{out}} \) expresses the time instant when the signaling message that was the response for the \( i \)th incoming signaling message, left the router.

**4.5.2 Average Traffic Forwarding Time**

The traffic forwarding time is the time that a data packet spends inside the router before it leaves the outgoing interface. This is a traditional performance metric for
router devices, but in this context, it also describes the effect of the resource reservation related tasks. The following formula gives the **average traffic forwarding time** \( t_D \):

\[
t_D = \frac{1}{n} \sum_{i=1}^{n} \frac{d_{i}^{\text{out}} - d_{i}^{\text{in}}}{n},
\]

where \( n \) denotes the number of the measurements, \( d_{i}^{\text{in}} \) expresses the time instant when the \( i \)th data packet entered to the router and \( d_{i}^{\text{out}} \) stands for the time instant when the data packet left the router.

### 4.6 Router Load Analysis

Preliminary benchmarking measurements have proven my assumption that the load on the router seriously affects the performance characteristics. For this reason, the proposed performance analysis framework characterizes the resource reservation capable routers in the function of the router load. But first, in order to determine how the router load impacts the reservation handling and data forwarding performance, I have analyzed the router load.

Generally, there is a terminology, in which the router load is associated to the utilization of the central processing unit. In this case this measure tells about the processing capabilities of the router that are already allocated for certain tasks. However, the central processing unit utilization does not reflect how the processing power is distributed among the tasks that the router performs. The assumption that the processing power is equally distributed among the competing tasks is not necessarily valid in most of the router products. Consequently, I have presumed that this router load description is not sufficient to describe the load for the performance characterization. A widely used alternative approach for the router load description is to use the number of reservation sessions as the most relevant parameter expressing the router load, such as in performance studies [42] and [43]. However, this approach neglects the traffic forwarding load. Since I have not agreed with this particular simplification, I have performed benchmarking measurements to confirm my assumptions that the router load cannot be specified by a single parameter.

In these measurements I have set up different load conditions on a Linux router running the ISI RSVP protocol by changing the number of sessions and the parameters of the data and signaling traffic. I have measured the utilization of the central processing unit and the two performance metrics: the signaling message processing time on the Path messages and the traffic forwarding time. In the measurements, the size of the data packets was 1024 bytes, and the rate of the data traffic was 3000, 4000 and 3250 packets per second in the order of the measurements. In the case of the last measurement there was an additional signaling flow going through the router, carrying 100 signaling messages in a second. Table 5 shows the result of the measurements.
The measurement results have proven that the load on the router significantly affects both the data forwarding and resource reservation handling performance of the routers. At the first two rows of Table 5, the router load is the same, but the performance metrics show difference. Thus, I have revealed that the traditional router load metric, expressing the utilization of the central processing unit, is not sufficient to identify the load conditions. Moreover, based on the results of the last two measurements, I have shown that the resource reservation session number in itself is also an imperfect measure to express the router load.

Originated from the router model, I have developed a novel router load description that characterizes the load on the router according to the load of the building blocks that the router model is built up. I have analyzed each building block and identified the parameters that are able to influence the overall utilization of the central processing unit and so affect the performance of the whole router. In the following sections I have identified only the possible sources of the performance impacts, while the exact amount of effect of these identified parameters can be determined using benchmarking measurements.

4.6.1 The Load on the Traffic Forwarder Block

First, in the case of the traffic forwarder block, I have determined the parameters of the traffic flow that are the root of the performance impact. First, the size of the data packets determines the amount of work that the router has to perform when it transfers the packets between the incoming and outgoing interfaces. Naturally, the rate of the incoming packets also affects the amount of work of the traffic forwarding block. Since in a real situation the rate and the size of the data packets change frequently it would be difficult to describe their effects. For this reason I have used a simplification and substituted the real traffic with an equivalent traffic flow, in which the size of the data packets are identical and the rate of the traffic is constant for a longer interval.

In addition to the traffic parameters, the number of allocated reservation sessions is also able to influence the performance. In the case of IntServ soft-state and hard-state protocols the traffic identifier block determines which flows are enabled for the priority treatment. Since this requires the classification of the corresponding reservation sessions, the work that the traffic identifier block has to perform depends on the number of session entries that the router has.

Thus, I have concluded that in my performance analysis framework, originated from the traffic forwarder block, the number of allocated resource reservation sessions ($r$),
the size \((s)\) and the forwarding rate \((q_T)\) of the transferred packets are those parameters
that are able to affect the performance of the resource reservation capable router.

4.6.2 The Load on the Signaling Message Processing Block

I have also analyzed the signaling message processing block, consists of the session
lookup block, the resource management block and the signaling message initiation
block. So as with the traffic identification block, the load on the signaling message
processing block, and this way the load on the router depends on the number of
allocated resource reservation sessions due to the session lookup block. The more
reservation sessions the router allocates the more time is spent on hunting the
the corresponding state information that are necessary to process the actual message.

The tasks of the resource management block and the signaling message initiation
block are determined by the type of the actions that the actual signaling message
prescribes. Consequently, the type of the signaling message, determining the jobs that
the router has to perform, influences the performance characteristics of the resource
reservation capable routers. In fact, not just the type of the message but also the
parameters of the message influence the amount of activities that the router performs
during the message processing. Considering a signaling message that refers to a
multiparty resource reservation session, it theoretically requires more processing
power than a signaling message that has the same type but refers to a unicast resource
reservation session. In my performance analysis framework, I have not performed
characterization align with the different parameters for the same type of signaling
messages, since there are huge differences among the protocols regarding they treat
these parameters. Instead, if one investigates the performance characteristics of a
router that processes a signaling message with a certain parameter setting then he or
she can do it treating this specially parameterized signaling message as a separate type
of the signaling messages.

Similar to the case of the traffic forwarding block, not just the effect of a single
signaling message is considered, but also the influence of the whole signaling
message flow. Thus, the rate of the signaling messages is also a parameter of the
router load. Again, for the sake of simplicity, in my performance analysis framework I
have considered constant rate signaling message flows only. Since the performance
metrics describe average values, this is an acceptable limitation of the model.

Over all, using the resource reservation capable model I have derived that due to the
signaling message processing block, the performance of the resource reservation
capable router depends on the number of reserved sessions \((r)\); and the incoming rate
\((q_S)\) and the type \((t)\) of the of signaling messages.

4.6.3 The Load on the Session Maintenance Block

At last, I have examined the session maintenance block. The duty of the session
maintenance block is to refresh the reservation sessions that the router has already
accepted. The resource reservation capable routers that initiate refresh messages
themselves independently of the incoming signaling messages, have to maintain at
least a list about the reservation sessions and their expiration time. The session
maintenance task selects the reservation sessions from the list that should be refreshed
and performs actions to refresh them. Both the session selection procedure and the
frequency of the refreshes depend on the number of the allocated resource reservation
sessions. This way, the number of resource reservation sessions \((r)\) influences the
performance of the session maintenance block and as a result it has impact on the overall performance of the resource reservation capable router.

4.6.4 The Parameters of the Router Load

During the construction of the resource reservation capable router model, I have assumed that the router does not perform any other relevant tasks, but the three main ones. Consequently, parameters that affect the performance of the tasks are the ones that specify the load conditions in the router. These five parameters are: the signaling flow parameters, which are the type of the signaling messages and the rate of the incoming messages; the data traffic parameters that are the size and the rate of the incoming data packets; and finally the number of resource reservation sessions.

In the case of routers where some of the building blocks do not exist due to the capabilities of the resource reservation protocol, it is expected then that the corresponding load parameters have no impact on the performance. For example, stateless protocols, where the resource reservation sessions are not recognized by the router, the session number parameter has no effect on the performance of the router.

4.7 Performance Characteristic Equations Based on the Router Model

Using the parameters of the router load I have established performance characteristic equation profiles expressing the performance metrics in a given load conditions. Although the particular performance characteristic equations differ from routers to router, they have common profile, since they are calculated using the same router model. In the suggested performance analysis framework, I have given the format of the performance characteristic equations only, while the exact performance characteristics that describe a resource reservation capable router, can be determined using customized benchmarking measurements detailed later.

4.7.1 Processing Power Scheduling in the Router Model

The proposed resource reservation capable router model specifies competition for common resources, such as the processing power, among the key building blocks. In my performance analysis framework I have described this competition based on an assumption that the activities of the building blocks are interrupted by other tasks, controlled by a scheduling algorithm. In order to control the access to the common resources, the scheduler associates a certain priority level to each task. Considering that the measured task competes for the processing power with a parallel task, there are three different possibilities derived from the priority rankings:

1. Assuming that the measured task has higher priority than the rival task, then there are no interruptions in the measured task at all, so the parallel task does not have impact on the measured activity. Using benchmarking measurements, this situation can be identified as the measured activity does not get influenced when there are different amount of work processed by the competing task and consequently there is no delay in the measured activity:

   \[d = 0.\]

2. The second situation is when the measured task has lower priority than the rival task. In this case there are as many interruptions in the measured activity
as many times the rival task requires processing. The length of the interruptions equals to the processing time of the competing task plus some overhead for each task switching. To identify this situation, one has to observe the rival task and if it does not slow down due to the activity of the measured task, then the measured task has lower priority than the competing one. If the activity of the competing task cannot be measured for some reason then the delay of the measured task also classifies this priority order. All the events of the higher priority task cause interruptions in the measured activity and therefore the delay of the measured activity can be computed as the number of events multiplied by the length of the interruptions. When the events arise periodically then the number of the interruptions can be calculated also from the rate of the events and the length of the measured activity. In this case the delay is described with the following formula:

\[ d = t_m \cdot q \cdot d_i, \]

where \( t_m \) is the measured length of the observed activity, \( q \) is the rate of the higher priority events and \( d_i \) is the delay that one interruption generates in the measured activity including the task switching time.

3. The final possibility is that the two tasks have the same priority. In this case the scheduling algorithm allocates fix time frames for the competing tasks equally and let them run in their time frame only. Since the scheduler run other processes parallel with the measured task, it causes delay in the measured activity. The amount of the delay equals to the time while the scheduling algorithm allocates the processing unit to the rival task. However, assuming a fair scheduler distributing the processing power equally between two processes having the same priority then the delay can be computed as the uninterrupted time of the measured process multiplied by a constant value that depends on the amount of work that the rival task performs:

\[ d = t_{m0} \cdot c, \]

where \( t_{m0} \) is the uninterrupted length of the measured activity and \( c \) is the value that express the effect of the parallel task.

Although processes usually have the same priority during their lifetime, it is also possible that certain functions performed by the tasks have different priorities. For example, processes often call system routines that are the part of the operating systems and perform input and output activities on the peripheries, such as the packet receiving on a network card. Since delayed input or output operations might cause information losses, these system calls are treated as high priority uninterruptible activities. Thus, when two tasks are competing for the processing power then using a system call, the lower priority task is still able to delay the higher priority process for the time of the system call.

Since there are no more priority orders between two competing processes that could be taken into account, and since all the different priority relationships have different impact on the measured activity, using the benchmarking measurement results, I have been able to identify the priority ranking of processes. Moreover, I have known the type of the functions describing the delay caused by the scheduling and thus I can approximate them accurately.
The scheduling model that I have sketched is known as the priority scheduling algorithm [50]. Its variants (e.g. Round-Robin scheduler) are the most common schedulers used in systems where there is a competition for common resources. In my performance analysis framework I have assumed that the competing tasks are scheduled based on a priority order, the same way as I described it.

As an example, Figure 48 presents the effect of the processing power schedule. In this imaginary scenario the signaling message processing and data forwarding tasks are interrupted several times while they finish the processing of the signaling message ($t_s$) and the forwarding of the data packet ($t_D$).

![Figure 48 - Possible Task Scheduling in a Resource Reservation Capable Router](image)

### 4.7.2 The Performance Characteristic Equation Profile of the Average Signaling Message Processing Time Metric

Based on the proposed resource reservation capable router model, the signaling messages pass through the signaling message processor only. Thus, the signaling message processing time depends on the time that the router allocates to process the message and the time that the scheduling algorithm allocates to perform other tasks while the signaling message is still being processed. Considering a signaling message processing that happens the same way as Figure 48 indicates, the interruptions of other tasks can be grouped by tasks. Figure 49 shows the grouped interruptions that delay the signaling message processing.
Relying on the scheduling model, I have given the profile of the signaling performance characteristic equation for the message processing time metric with the following equation:

\[ t_s = t_{s0}(r,m) + d_{SD}(t_{s0},t_{D0},q_T,t_s) + d_{SM}(t_{s0},t_{M0},r,t_s), \]

where \( t_{s0}(r,m) \) is base signaling message processing time, the time that the processing of a certain type \((m)\) signaling message takes in a router that allocates a given number of sessions \((r)\) and perform no other tasks during the signaling message processing. I have used the \( d_{SD}(t_{s0},t_{D0},q_T,t_s) \) function to express the delay caused by the interruptions of the data traffic forwarder block. Here I have specified all the parameters that are able to influence this delay through the priority scheduling algorithm. If the traffic forwarder block has higher priority than the signaling message processing block has then the delay depends on the length of the uninterrupted data forwarding time and the number of the interruptions that is affected by the rate of the incoming data packets \((q_T)\) and the length of the signaling message processing \((t_s)\). In that case when the two tasks have the same priority then the base signaling message processing time \((t_{s0})\) also affects the length of the delay. Similarly, I have used function \( d_{SM}(t_{s0},t_{M0},r,t_s) \) to represent the delay caused by the interruptions of the session maintenance block. Here, in the case, when the session maintenance is higher prioritized than the signaling message processing task then the length of the interruptions depend on the time while a single sessions is maintained uninterrupted \((t_{M0})\) and the number of interruptions is determined using the number of allocated sessions \((r)\) and the length of the signaling message processing \((t_s)\). If the two tasks have the same priority then the base signaling message processing time \((t_{s0})\) also has impact on the delay. In the case, when the signaling message processing block bears higher priority than one of the other tasks, then that task should not interrupt the signaling message processing. In this case, the benchmarking measurements should indicate that the corresponding \( d_{SD} \) or \( d_{SM} \) function has no impact on the performance.

### 4.7.3 The Performance Characteristic Equation Profile of the Average Traffic Forwarding Time Metric

The shape of the performance characteristic equation profile that describe the average traffic forwarding time metric is very similar to the previous performance characteristic equation profile, since both performance characteristic equation comes from the same model, assuming interruptions generated by other tasks.
In the proposed router model data packets cross through the traffic forwarder block that routes, forwards and therefore necessarily delays them. So as with the signaling message processing time characteristic equation profile, the other two tasks also delay the traffic forwarding. The following equation gives the traffic forwarding time performance metric:

\[ t_D = t_{D0}(r,s) + d_{DS}(t_{D0},t_{S0},q_S,t_D) + d_{DM}(t_{D0},t_{M0},r,t_D), \]

where \( t_{D0}(r,s) \) is the base traffic forwarding time, the time that a data packet, with a given length \((s)\), spends inside a router that allocates a number of resource reservation sessions \((r)\), but performs no other tasks then data forwarding. The \( d_{DS}(t_{D0},t_{S0},q_S,t_D) \) function expresses the delay caused by the interruptions of the signaling message processing task. Considering that the priority of the signaling message processing task is higher than the priority of the traffic forwarding task, then this delay depends on the uninterrupted signaling message processing time \((t_{S0})\), the rate of the incoming signaling messages \((q_S)\) and the length of the traffic forwarding time \((t_D)\). In the case of equal priorities the delay depends on the base traffic forwarding time \((t_{D0})\). Similarly, the \( d_{DM}(t_{D0},t_{M0},r,t_D) \) function tells the delay due to the interruptions of the session maintenance task. The parameters are the uninterrupted processing time of one session refresh \((t_{M0})\), the number of sessions \((r)\) and the total traffic forwarding time \((t_D)\) for the case when the session maintenance is prioritized over the traffic forwarding task. If the priorities of the two tasks are the same then the delay also depends on the base traffic forwarding time \((t_{D0})\). When the priority level of the traffic forwarding task is higher than any of the other tasks then the benchmarking measurement should result that the corresponding \( d_{DS} \) or \( d_{DM} \) are zero, due to the lack of interruptions.

Although the given performance equation profiles are too general, I have specified the most dominant parameters that are able to influence the performance metrics. Using benchmarking measurement the performance equation profiles can be determined for individual resource reservation capable routers. In these performance characteristics only those parameters remain that have impact on the performance metric.
5 Benchmarking Resource Reservation Capable Routers

Besides the profiles of the performance characteristic equations the performance analysis framework also needs a benchmarking measurement methodology that can be used on an arbitrary router product to determine its exact performance characteristics. This measurement methodology consists of customized benchmarking measurements that are constructed in a smart way to focus on a specific function of the performance characteristic equation profile and to exclude the unknown effects of the other building blocks.

In addition to the presented performance analysis framework I have also tested it on two resource reservation capable routers. The special benchmarking methodology is also presented using these routers as examples. Moreover based on the calculated performance characteristics of these resource reservation capable routers I have also validated the performance analysis framework and my assumptions about the router model. To demonstrate the performance analysis framework I have selected the following router products and resource reservation protocol implementations:

- ISI implementation of the RSVP protocol running on a Linux based PC router
- Boomerang protocol running on a Linux based PC router

I have chosen Linux based PC routers, since this way I had a chance to look inside the code of the router and eventually I was also able to place checkpoints inside the router and measure the internal mechanisms. Linux based PC routers are not used in real networks, but according to my measurement experiences, in most cases, especially in the case of reservation handling, they behave the same way as dedicated router products. Nevertheless, the investigated resource reservation protocol implementations are just two examples from the large variety of resource reservation protocol implementations, but these two protocols have lots of differences in their attributes and therefore cover a larger set of the resource reservation protocols. Both protocols maintain soft-states, but while Boomerang is an instant response protocol with sender orientation then the RSVP protocol is receiver oriented protocol requiring two passes for the reservation setup. Necessarily RSVP also needs extra state lookups and generates signaling messages itself. In addition, while the architecture of the router is the same, the implementation also significantly differs. The RSVP protocol runs in the user-space, while Boomerang runs in the kernel-space. Because of the kernel-space code is closer to the operating systems therefore this implementation acquires faster system calls and shorter task switching times.

5.1 Measurement Methodology

In order to measure the signaling message processing and data forwarding time performance metrics, I have captured all the data and signaling packets that appeared on the network interfaces of the router during the measurement. This way, after the measurements from the recorded log file that contained the capturing time and the type of all the packets I have calculated the investigated performance metrics. Since the packet capturing introduces negligible delay according to my measurements, the performance metrics are not biased seriously. For the same reason I recommend to use this method for all the benchmarking measurements.
5.1.1 Benchmarking Measurement Scenario

Figure 50 presents a benchmarking measurement scenario, where the resource reservation is performed by the Router connecting Tester 1 and Tester 2. Signaling messages on ingress and egress ports of the router are intercepted and logged along with time-stamps by the Sniffer computer that runs a packet capturing software and is connected to both sides of the router via network hubs. This way all the data packets and signaling messages on the network interfaces of the router are captured. I have suggested using network hubs, since they have almost negligible forwarding delay and thus they do not affect the measurement results.

![Figure 50 - Benchmarking Measurement Scenario](image)

Although the network hubs delay the network packets almost negligibly, in the case of PC routers even these elements can be removed from the configuration to obtain more accurate measurements. Figure 51 shows a benchmarking scenario that I have designed for PC routers and PC tester devices. In this case, there are four testers, from which two are responsible for the signaling flows and the other two generate the data traffic. The four PC testers are able to provide enough processing power that the generation of the data and signaling traffic requires. The packet capturing code is built in the tested PC router device using fast kernel routines. The fast kernel code assures that the packet capture runs as the highest priority task and captures the packets immediately as they appear on the network card. I have measured that the accuracy of the capturing module is on a nanosecond order. I have also investigated the delay that the packet capturing module causes in the performance of the router using external measurements. The measurements revealed that the existence of the packet capture module does not limit the performance of the tested router measurably.

![Figure 51 - Benchmarking Measurement Scenario for PC Based Routers](image)

The only delay that cannot be detected in this measurement scenario is the time that is required by the network interface to grab the packet from the physical line. Still, I have estimated this delay by knowing the line speed and the length of the packets. From the calculation I have got that this delay, which is around 80 nanoseconds for a
1000 bytes packet on a 100 Mbps link, is negligible compared to the magnitude of the benchmarking results.

Using PC routers running Linux operating system I had the chance to place benchmarking and profiling checkpoints into the code of the router and resource reservation protocol implementations. Using profiling measurements I have measured the time that a certain building block of the resource reservation protocol takes. For this purpose I have developed a Linux kernel module capturing internal system events and time-stamping them using the real-time clock with nanosecond accuracy. I have set up hooks to trigger my kernel module (i) when a data packet or a signaling message appears on the network interface; (ii) when the packet is transferred from the network card to the operating system; (iii) and also in special occasions when a special code in the routing or the signaling message processing signals it. Using the timestamps that indicated the time when the packet was transferred to the operating system I have excluded the effect of packet queuing in the network interfaces that would scatter the benchmarking results seriously. Timestamps on special codes have provided the possibility to place several checkpoints into the code of the resource reservation protocol implementation and profile the internal algorithms.

According to the general benchmarking method, during the measurements the testers have set up the required load conditions on the tested router using several data traffic and signaling message flows. After a short time, when the router was already over its transient behavior caused by the appearance of load, one of the testers initiated a test flow constructed from a sequence of packets that the router has recorded. The flows consisted of thousand of packets, since the accurate results require large numbers of measurements. The packets and signaling messages in the test flow were arranged in a way that they have not changed the load conditions on the router. Thus, each packet perceived the same treatment as the others. Due to the limited event storage space of the packet capture module, the module has captured only the packets of the test flow and recorded no other packets. After the measurements the tested router and the testers have been reset to their original conditions.

The benchmarking measurement configuration, in which I have performed the benchmarking measurements, was the same as it can be seen in Figure 51. The tested router was a PC with a Pentium II 400 MHz processor, 128 megabyte memory and two double port Intel EtherExpress Pro cards. It ran a 2.2.14 Linux kernel from the Debian distribution [51]. The applied resource reservation protocols were the latest releases of the ISI RSVP and Boomerang protocols. The tester computers had the same hardware and software configurations as the router.

5.2 Determining the Exact Performance Characteristic Equations

The goal of the benchmarking measurements was to determine the exact performance characteristics that describe the tested router. Previously, based on the resource reservation capable model, I have specified the performance characteristic equation profiles expressing the performance metrics. These two performance characteristic equation profiles are consisted of six performance characteristic functions, from which two describe the base value of the performance metric and other four expresses the delay that the competing tasks generate. Using the benchmarking measurements I have derived the performance characteristic functions and finally I have given the
exact performance characteristics of the tested resource reservation capable routers. The two performance characteristic equation profiles are:

\[ t_S = t_{S_0}(r,m) + d_{SD}(t_{S_0}, t_{D0}, q_F, t_S) + d_{SM}(t_{S_0}, t_{M0}, r, t_S). \]

\[ t_D = t_{D_0}(r,s) + d_{DS}(t_{D_0}, t_{S0}, q_S, t_D) + d_{DM}(t_{D_0}, t_{M0}, r, t_D). \]

During the investigations I have performed several measurements to characterize a certain performance characteristic function.

### 5.2.1 The Base Signaling Message Processing Time: \( t_{S_0}(r,m) \)

First, I have explored the base signaling message processing time, \( t_{S_0}(r,m) \) that is the time that the signaling message processing task exclusively spends with the processing of the signaling message.

In order to determine the amount of time that is required by the signaling message processing task, I have constructed a measurement where there were no other tasks running on the tested router, but the signaling message processing task. Since the base signaling message processing time depends on the number of resource reservation sessions and the type of the signaling message, I have measured it in the function of these two parameters. However, when a soft-state router allocates resource reservation sessions it also maintains them running the session maintenance task in parallel with the signaling message processing task. In order to stop the session maintenance task I have used extremely long soft-state refresh intervals defined in the signaling messages. The sequence of the signaling messages in the test flow was also special. I have used alternating state setup and teardown messages in the flow to avoid significant and permanent load changes in the tested router. The timing of the signaling messages was also special. I have observed that equally spaced signaling messages might cause the synchronization of the different tasks in the resource reservation capable routers and so it spoils the measurement results. To prevent the synchronization I have used random timing between two signaling messages in the flow. Also, the time between two messages was long enough to avoid the queuing of the signaling messages. The exact measurement methodology that I have applied to characterize the base signaling message processing time \( t_{S_0}(r,m) \) is the following:

*I have benchmarked the signaling message processing time of signaling messages in a test flow. The flow has consisted of a certain type of signaling messages that followed each other with a long and unpredictable break. In the case of the reservation setup or teardown type messages, I have always used them in pairs alternating in the test flow and thus I have avoided the change of the load conditions in the tested router. Throughout the measurement, the router was loaded by various amounts of resource reservation sessions, but these allocations were set up specifying a long refresh interval, consequently requiring no refresh messages or other session maintenance actions. During the measurements there was no data traffic forwarded by the router.*

Figure 52 presents the signaling message processing time measurement results in the function of the allocated resource reservation session number for the Boomerang resource reservation protocol. This protocol has only one signaling message type that signals both the reservation setup and reservation teardown. The two functions differ in the value of the resource parameter used in the message.
Based on the measurement results I have determined that in this case the signaling message processing time depends on the number of allocated resource reservation sessions. To approximate the signaling message processing time I have fitted a regression formula to the measurement samples. The best fitting formula based on the maximal R-squared value\(^1\) [52] is achieved by a power trend line:

\[
t_{39}(r) = 42\mu s + 0.1374\mu s \cdot r^{0.3966},
\]

where \(r\) is the number of allocated resource reservation sessions.

In the case of the ISI RSVP protocol there are five base signaling message primitives. The Path message sets up the path of the data traffic and the PathTear signaling message tears it down. Similarly, the resource reservation is initiated by the Resv signaling message and terminated by the ResvTear message. The fifth signaling message type is the Conf signaling message that acknowledges the successful resource reservation. Since the Conf message is not mandatory for the proper operation of the protocol, I have not investigated it. In the case of the Path and Resv messages there are certain parameters that describe the multiparty reservation handling scheme for routers. Previously I have shown that these reservation schemes require different amount of work during the signaling message processing. In this measurement I have selected the wildcard reservation style and constructed signaling messages that referred two signaling endpoints.

Figure 53 and Figure 54 illustrate the results of the benchmarking measurements in the function of the resource reservation session number, separated by the signaling message types.

---

\(^1\) R-Squared is also known as co-efficient of determination or Goodness of Fit value. It indicates how best least square regression line is fitted.
Based on the measurements results I can state that in a router that implements the ISI RSVP protocol the base signaling message processing time is influenced by the number of allocated resource reservation sessions. To give an approximation to the results I have used a second order polynomial type regression maximizing the R-squared value:

\[ t_{so}(r) = t_{r0} + t_{r1} \cdot r + t_{r2} \cdot r^2, \]

where \( r \) is the number of the allocated resource reservation sessions. The coefficients of the regression are \( t_{r0} \), \( t_{r1} \) and \( t_{r2} \) that all depend on the type of the signaling message. Table 6 shows these parameters in the case of the most important four signaling message types.

Table 6 - Coefficients for the Base Signaling Message Processing Time Approximation in the case of the ISI RSVP Protocol

<table>
<thead>
<tr>
<th></th>
<th>PATH [\mu s]</th>
<th>RESV [\mu s]</th>
<th>PATH TEAR [\mu s]</th>
<th>RESV TEAR [\mu s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{r0} )</td>
<td>273.1</td>
<td>822</td>
<td>131.5</td>
<td>412.6</td>
</tr>
<tr>
<td>( t_{r1} )</td>
<td>0.1573</td>
<td>0.1687</td>
<td>0.0019</td>
<td>0.1066</td>
</tr>
<tr>
<td>( t_{r2} )</td>
<td>0.000017</td>
<td>0.000017</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

During the benchmarking tests I have increased the number of sessions up to 1500 allocations only, since the Linux router I have used for the measurements was not able to allocate more sessions in the traffic control module of the operating system.

In spite the benchmarking measurements were sufficient to determine the exact characteristics of the base signaling message processing time, I have performed additional profiling measurements to examine the results. I have looked for the reason for the huge difference between the measured base signaling message processing time metrics in the case of the two different protocols. First, I have examined the session lookup building block. Figure 55 presents the session lookup time in the function of the allocated sessions for both resource reservation protocols. In this case I have separated the code of the session lookup algorithm from the source code of the protocols and I have driven them directly assuring to achieve extremely large number of sessions.
This measurement has confirmed that in the case of the two protocols the session lookup building block is the root of the session number dependency. In the case of the ISI RSVP protocol, signaling messages require not only session lookups but eventually to search for other states as well, multiplying the delay that this measurement indicates.

Other profiling measurements have highlighted that in the case of the ISI RSVP protocol the reservation management and signaling message forwarding block take considerable large time. In the case of the Path message the route query block takes about 60 microseconds and the signaling message forwarding is up to 150 microseconds, which is the majority of the whole base signaling message processing time. In the case of the Resv message the reservation dedication block allocates the processor for about 630 microseconds, while the signaling message forwarding is an additional 90 microseconds. These values seems to be extremely high compared to the Boomerang protocol that performs the whole base signaling message processing under 50 microseconds in the investigated session number range. The reason for the difference is the dissimilarity of the implementations and the low complexity of the Boomerang protocol.

5.2.2 The Base Traffic Forwarding Time: $t_{D0}(r,s)$

I have determined the exact characteristics of the base traffic forwarding time $t_{D0}(r,s)$, in the case of the two investigated resource reservation capable routers. The base traffic forwarding time represents the pure time that the scheduler allocates to the traffic forwarder block to forward a data packet. In my resource reservation capable router model, the traffic forwarder block consists of concatenated traffic identification and forwarding blocks. In the model I have assumed that the traffic identification is sensitive to the number of resource reservation sessions, while the traffic routing and forwarding block is influenced by the size of the data packets. Since these two blocks do not depend on the load parameter of the other block, I have analyzed their effects independently:

$$t_{D0}(r,s) = t_{D0S}(s) + d_{D0R}(r),$$

where $t_{D0S}(s)$ function gives that part of the base traffic forwarding time that depends on the packet size, while the $d_{D0R}(r)$ function describes the delay that the allocated resource reservation sessions cause to the data traffic forwarding.

The benchmarking methodology is the following:
I have characterized the base traffic forwarding time using two benchmarking measurement sequences. In the first measurement I have sent a data flow through the tested router where I have changed the size of the packets. This measurement has shown how the base traffic forwarding time depends on the size of the packets. In a second measurement I have set up various number of resource reservation allocations in the tested router. I have set extremely long refresh period times to exclude the effects of the refresh messages and to neglect the activity of the session maintenance block. To measure how the number of allocated sessions affects the base traffic forwarding time I have sent a data flow through the router and repeated the measurements changing the number of allocated sessions. Finally, I have combined the results into a single performance characterization function.

To quantify the \textit{base traffic forwarding time} dependence on the packet size $t_{\text{DOS}}(s)$, I have performed measurements using traffic flows with different packets sizes. Since I have performed measurements using an Ethernet networking card, I have used a packet size in the tests that were smaller than the fragmentation limit and greater than the minimal required length specified in the standards. Figure 56 shows the measured traffic forwarding time, when there is no reservation session allocated in the router:

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{traffic_forwarding_time.png}
\caption{Linux Router, Traffic Forwarding Time vs. Packet Size}
\end{figure}

The curve of the $t_{\text{DOS}}(s)$ function looks like that it is a concatenation of two functions. If the packet size is below 256 bytes then the data traffic forwarding shows a linearly increasing trend in the functions of the packet size, while packets longer than 256 bytes are forwarded with a constant speed, and the traffic forwarding time is 14.6 $\mu$s. The following equation gives the base traffic forwarding time in the function of the size of the packets:

$$
t_{\text{DOS}}(s) = \begin{cases} 
12.263 \mu s + 0.0093 \left( \frac{s}{\text{byte}} \right) \cdot s & |s < 256 \text{ byte} \\
14.6 \mu s & |s \geq 256 \text{ byte}
\end{cases}
$$

where $s$ is the size of the packet.

In the case of the previous benchmarking measurement, both resource reservation capable routers behave the same way and I have got identical results. The reason is that the routing and forwarding block is the part of the Linux operating system and therefore it was exactly the same for both measurements.

However, in the following measurements, where I have tested how the base traffic forwarding time depends on the number of allocated sessions, the traffic identification block were different in the case of the analyzed resource reservation capable routers. In the case of the router using the ISI RSVP protocol, the traffic identification was
performed by the Traffic Control module that is the part of the Linux operating system. In the case of the Boomerang protocol, it is also the Boomerang resource reservation protocol implementation that performs the traffic identification.

Figure 57 and Figure 58 show the results of the benchmarking measurements that investigate the delay on the base traffic forwarding time caused by the allocated resource reservation sessions. Here, I have measured the total forwarding time and get the presented delay by subtracting the amount of time that the routing and forwarding requires.

In the case of the Boomerang protocol, I have used a regression formula maximizing the R-squared value, based on a power trend line to approximate the base traffic forwarding delay that depends on the allocated number of resource reservation sessions:

\[ d_{DOR}(r) = 0.692 \mu s \cdot r^{0.2283}, \]

where \( r \) is the number of resource reservation sessions.

In the case of the ISI RSVP implementation that uses the built in Linux Traffic Control module, I have used a linear regression maximizing the R-squared value to guess the allocated resource reservation session number dependent base traffic forwarding delay curve:

\[ d_{DOR}(r) = 0.2 \mu s + 0.0005 \mu s \cdot r, \]

where the number of resource reservation sessions is denoted by \( r \).

In Table 7 I have summarized the base traffic forwarding time by accumulating the packet size dependent base traffic forwarding time and the delay that caused by the resource reservation sessions:
Table 7 - Base Traffic Forwarding Time for Small Packets

<table>
<thead>
<tr>
<th>Protocol</th>
<th>( t_{do}(r,s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boomerang</td>
<td>( 12.263 \mu s \cdot 0.0093(\frac{\mu s}{\text{byte}}) \cdot s + 0.692 \mu s \cdot r^{0.2283} )</td>
</tr>
<tr>
<td>ISI RSVP</td>
<td>( 12.263 \mu s \cdot 0.0093(\frac{\mu s}{\text{byte}}) \cdot s + 0.2 \mu s + 0.0005 \mu s \cdot r )</td>
</tr>
</tbody>
</table>

Table 8 - Base Traffic Forwarding Time for Large Packets

<table>
<thead>
<tr>
<th>Protocol</th>
<th>( t_{do}(r,s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boomerang</td>
<td>( 14.6 \mu s + 0.692 \mu s \cdot r^{0.2283} )</td>
</tr>
<tr>
<td>ISI RSVP</td>
<td>( 14.8 \mu s + 0.0005 \mu s \cdot r )</td>
</tr>
</tbody>
</table>

5.2.3 The Signaling Message Processing Delay Caused by the Data Forwarding Task: \( d_{sd}(t_{s0},t_{do},q_{T},t_{S}) \)

I have determined the delay of the signaling message processing activity caused by the interruptions of the data forwarding task for the two investigated protocols. In the performance characteristic equation profile I have identified that this delay depends on the length of the interruptions that is affected by the base traffic forwarding time; the number of interruptions that is influenced by the length of the signaling message processing; the incoming rate of the data packets; and the base signaling message processing time.

During the measurements I have investigated the signaling message processing time in the function of the data traffic that went through the router simultaneously. I have performed several benchmarking measurements using various rates for the data packets in the generated traffic flow. The exact measurement methodology is the following:

I have set up a data traffic flow using constant sized and equally spaced packets. The spacing between the data packets determines the rate of the packets. During the measurements I have constructed different traffic flows using various packet rate parameters. I have also generated a signaling flow from signaling messages that followed each other with a long and unpredictable break. The signaling flow was constructed similarly as in the case of the base signaling message time measurement, so signaling messages influencing the load in the router were placed in pairs into the flow. In the measurements the tested router forwarded the data traffic to its destination and also processed the signaling messages at the same time. I have measured the processing time of every signaling message. Since this delay is not affected by the allocated resource reservation directly, I have not set up any reservations during the tests.

In order to get the signaling message processing delay caused by the data forwarding task I have subtracted the base signaling message processing time values from the results of the measurements. From these delay values, using the functions that describe the shape of the delay in the case of various priority rankings between the signaling message processing and data traffic forwarding tasks, I have deduced the actual priority order of the tasks. This way I have also got the shape of the function that approximates the delay.

Figure 59 represent the measurement results obtained on a Linux router running the Boomerang resource reservation protocol.
The results indicate that the signaling message processing time depends on the rate of the data packets, thus the whole signaling message processing task does not have higher priority than the data traffic forwarding task. Assuming that the data traffic forwarding task has higher priority than the signaling message processing task then it is the incoming packets that generates the interruptions. In this case the length of the interruptions is the length of the base data traffic forwarding activity plus some overhead for the task switches. Fitting the formula that approximates the delay in the case of this priority order, I have got that its curve matches the measurement results. Figure 60 shows the interruption lengths in the function of the data traffic rate, assuming that the traffic forwarding task has higher priority than the signaling message processing task. Since this delay does not depend on the rate of the data packets, I have shown that in the case of the Boomerang protocol implementation, the traffic forwarding task has higher priority than the signaling message processing task. The delay that caused by the interruptions of the data traffic forwarding is approximated with the following equation:

\[ d_{SD}(t_{D0}, q_T, t_s) = t_s \cdot q_T \cdot (t_{D0} + d_{SW}), \]

where \( t_s \) is the total signaling message processing time, \( q_T \) denotes the rate of the data packets, \( t_{D0} \) is the base traffic forwarding time and finally \( d_{SW} \) expresses the time that a task switch takes.

From the measurement results I have computed that each task switching action requires 27.8 \( \mu s \) in the case of the Boomerang protocol.

Figure 61 and Figure 62 show the measurement results for the investigated signaling messages in the case of the ISI RSVP protocol.
Based on the calculations about the signaling message processing delay results, I have determined that similarly to the Boomerang protocol, in the case of the ISI RSVP implementation the data traffic forwarding task has higher priority than the priority of the signaling message processing task. Due to the similarity, I have used the same equation to approximate the examined delay:

\[
d_{SD}(t_{D0}, q_T, t_S) = t_S \cdot q_T \cdot (t_{D0} + d_{SW}),
\]

where \(t_S\) is the total signaling message processing time, \(q_T\) is the rate of the data packets, \(t_{D0}\) is the base traffic forwarding time and \(d_{SW}\) denotes the time that a task switch takes.

Figure 63 presents the calculated interruption length values according to the data traffic rate in the case of the RSVP messages.

Despite that the Boomerang protocol runs on the same Linux router as the ISI RSVP protocol, I have obtained different values for the task switching delays. In the case of the ISI RSVP implementation, each task switch takes 40.3 µs. The result is not surprising, since the two resource reservation protocol implementations run in different code spaces. The Boomerang protocol is a kernel module and therefore the task switches are faster between two kernel level codes than in the case of the ISI RSVP that runs as a user level code.

As a validation of my assumption about the priority of the processes, I have performed measurements, where I have recorded the number of data packets that left the router during the signaling message processing. The measurement results have revealed that all the data packets that arrived during the signaling message processing
activity left the tested router. This proves that signaling message processing task was interrupted for the length of the data packet forwarding in the case of each packet. Thus, the results of this measurement revealed that the data traffic forwarding has higher priority than the signaling message processing has in the case of both resource reservation protocol implementations.

During the measurements I have made my investigations as the function of the data traffic rate parameter that never exceeded the 7000 packets per second limit. The reason for this limitation is that the traffic generator utility was not able to generate more intensive data traffic. However, this traffic rate, considering 1024 bytes long packets, that I have used for the measurements equals to a 55 Mbps traffic volume, which volume is close to the total utilization of a 100 Mbps Ethernet link.

5.2.4 The Signaling Message Processing Delay Caused by the Session Maintenance Task: $d_{SM}(t_{S0}, t_{M0}, r, t_S)$

In order to complete the exact performance characteristic equation that describes the signaling message processing time metric, I have identified the signaling message processing delay that is due to the interruptions of the session maintenance task. The examination method is analogue to the signaling message processing delay investigation when it depends on the interruptions of the data forwarding block. Based on the benchmarking measurement results I have identified the actual priority order between the two tasks and then I have approximated the delay curve.

Here, I have presumed that in the case when the session maintenance task has higher priority than the signaling message processing task then the length of the interruptions is affected by the time that one session maintenance activity requires, while the number of the interruptions depends on how many times the router initiates session maintenance actions during the signaling message processing activity. Considering the refresh signaling messages, the session maintenance block is utilized each time a refresh message arrives or leaves the router. In the case of instant response resource reservation protocols, like the Boomerang protocol, the router does not initiate refresh messages by itself, just renews the states of the corresponding session and forwards the message along the reservation path. In contrast, non-instant response protocols, such as RSVP, schedules the refresh message initiations according to its own timer. These routers drain the incoming refresh signaling messages refreshing the referred sessions and initiate new refresh messages independently of the arriving time of the incoming messages. Since in this case it is not the same code that handles the incoming and outgoing refresh signaling messages, I have examined their effects separately.

Firstly, I have considered the case when the active state of a resource reservation session expires and therefore the resource reservation protocol initiates a refresh signaling message refreshing the session in the neighboring routers. In this case the number of expired resource reservation sessions depends on the total number of resource reservation sessions. Secondly, I have considered the case when the router receives a refresh signaling message and this invokes the session maintenance block to renew the states of the corresponding resource reservation session. In this case the number of incoming refresh messages depends on the total number of resource reservation sessions. Although in both cases the delay depends on the total number of resource reservation sessions, the length of the interruptions and also the priority of the refresh handling routines might be different. In the case of non-instant response
resource reservation capable routers I have derived the following equation to estimate the delay caused by the interruptions of the session maintenance task:

\[ d_{SM} \left( t_{S0}, t_{M0}, r, t_S \right) = d_{SMI} \left( t_{S0}, t_{M0}, r, t_S \right) + d_{SMO} \left( t_{S0}, t_{M0}, r, t_S \right), \]

where the \( d_{SM}(t_{S0},t_{M0},t_S) \) function describes the influence of the incoming refresh signaling messages and the \( d_{SMO}(t_{S0},t_{M0},t_S) \) function express the effect of the outgoing refresh message initiations. The parameters are the same as the parameters of the \( d_{SM} \) function: \( t_{S0} \) is the base signaling message processing time, \( t_{M0} \) is the time that one session maintenance activity takes, \( r \) is the number of allocated sessions and \( t_S \) is the total signaling message processing time.

In the case of the instant response resource reservation capable routers the session maintenance activity includes the receiving, processing and the forwarding actions for a certain refresh signaling message at the same time, therefore I have not distinguished these actions here.

Based on the instant response feature of the tested protocol the benchmarking methodology also differs:

In the case of instant response protocols I have set up a number of resource reservation sessions in the router. The signaling endpoints maintained the sessions sending refresh signaling messages using the timing that the protocol defines. I have constructed a signaling test flow in a similar way as in the case of the previous measurements, using alternating signaling message pairs if it was required. I have sent the test flow through the router and measured the signaling message processing time of each signaling message. I have made several benchmarking measurements in the parameter of the allocated session number. Finally, I have got the delays caused by the interruptions of the session maintenance task by subtracting the corresponding base signaling message processing values from the measurement results. During the tests there was no data traffic forwarded by the router.

In the case of non-instant response protocols, first I have characterized the delay caused by the initiated refresh signaling messages. I have set up a number of resource reservation session in the resource reservation capable router and constructed the test signaling message flow the same way as in the case of the instant response protocol measurement. However, just before sending the test flow through the router, I have cut off the refresh signaling message generation in the signaling endpoints. Thus, during the measurements there were no incoming refresh messages. The results of the benchmarking is the delay that I have got the same way as in the case of the instant response protocol, subtracting the base signaling message processing time. In order to identify the delay generated by the interruptions of the session maintenance task due to the incoming refresh signaling messages, I have used the same measurement method as it is described for the case of the non-instant response protocols, but this time I have got the investigated delay by subtracting the results of the previous measurements that gives that signaling message processing time, which is affected by the outgoing refresh signaling messages.

Examining the Boomerang implementation I have obtained that the signaling message processing time depends on the number of allocated resource reservation sessions, as it can be seen in Figure 64. This indicates that the signaling message processing task has not got higher priority than the session maintenance task has.
Observing the order of the signaling messages that leave the tested router I have noticed that in the case of a near simultaneous arrival, always the non refreshing signaling message is the first outgoing message regardless of the sequence of the arrival. Thus, I have concluded that the signaling message processing task still has higher priority than the session maintenance task, but some part of the session maintenance task calls certain kernel routines that interrupt the signaling message processing. I have suspected that when a refresh message arrives to the router then the router interrupts the actual performed task to transfer the incoming packet to the main memory in order to avoid packet losses. In this case the number of interruptions is calculated by multiplying the rate of the signaling messages with the total signaling message processing time, while the interruption length should be constant. The protocol describes that each session should be refreshed in each 30 seconds and therefore the intensity of the refresh signaling messages is also predictable. Fitting the appropriate formula, describing the interruptions of a higher priority task, to the measurement results (Figure 65), I have got that the interruption length is independent of the number of allocated sessions. Consequently, my assumption was right, so the signaling message processing task has higher priority than the session maintenance task, but even in this case, the signaling message processing is interrupted for 14.2 µs when a refresh signaling message arrives.

In summary, the following expression describes the delay that the session maintenance activity causes in the signaling message processing task:

\[ d_{SM}(t_{SM}, r, t_S) = t_S \cdot \frac{r}{30 \text{sec}} \cdot 14.2 \mu \text{s}, \]

where \( r \) is the number of allocates resource reservation sessions.

In the case of the ISI RVSP protocol, first I have investigated the \( d_{SMO} \) function that tells the amount of delay that the outgoing refresh signaling messages cause in the signaling message processing activity. The delay is generated by the session maintenance task that searches for the expired sessions and initiates refresh signaling messages in order to renew them in the neighboring routers. Figure 66 and Figure 67 show the calculated delay that is the measured signaling message processing time not including the base signaling message processing time. The results are represented in the function of the allocated resource reservation session number.
The measurement results revealed that the session maintenance activity refreshing the resource reservation sessions seriously influences the signaling message processing task. However, in the case of the Path and Resv messages the amount of the delay is nearly the same, consequently the length of the delay does not depend on the total signaling message processing time directly. Analyzing the measurement results I have observed that in the case of a certain number of allocated sessions, the results are concentrated around different levels of the delay. Figure 68 and Figure 69 show histograms of the signaling message processing delay values in the case of 1000 and 1500 allocated resource reservation sessions excluding the effect of the incoming refresh messages.

Examining the histograms I have found that the difference between two delay peaks is constant in the case of a given number of sessions. However, this difference increases as the router allocates more resource reservation sessions. Based on these facts I have assumed that each refresh signaling message generation adds a considerably long delay to the signaling message processing activity, and therefore I have investigated the session maintenance activity in the function of the generated refresh messages. Figure 70 show the length of one refresh message generation in the function of the allocated session number, while Figure 71 show the average number of outgoing refresh messages that leaves the router during one signaling message processing also in the function of the session number. The measured values do not depend on the type of the processed signaling message.
Based on these measurement results I have determined that the length of the refresh message generation does not depend on the total signaling message processing time. This indicates that the session maintenance task has higher priority than the signaling message processing task. However, in this case, the number of refresh signaling messages generated by the router should be influenced by the total signaling message processing time. Instead, this is not the case, but the number of refresh messages leaving the router is also independent of the total signaling message processing time. According to my assumptions, the session maintenance task has higher priority than the signaling message processing task has, but the number of the sessions to be refreshed is determined independently from the signaling message processing activity. This way I have used the following expressions to approximate the delay caused by the generation of the refresh signaling messages:

\[
d_{\text{ROUT}}(r) = 53.83 \mu s + 0.2852 \mu s \cdot r,
\]

\[
f_{\text{ROUT}}(r) = 0.00031 \cdot r,
\]

where the \(d_{\text{ROUT}}(r)\) function gives the length of one refresh message generation in the function of the allocated session number denoted by \(r\). Secondly, the \(f_{\text{ROUT}}(r)\) function describes the number of refresh message generation during a signaling message processing activity.

Besides the delay caused by the generation of the refresh signaling messages, there is also a delay that is based on the activity that takes care of the expired resource reservation sessions. This delay can be measured as the delay of those signaling message processing activities where there is no outgoing signaling message. Figure 72 and Figure 73 presents the result of these measurements.
Since the delay caused by the same session maintenance activity watching expired sessions is different for the various types of signaling messages, I have expected that the base signaling message processing time affects this delay. In this case, this part of the session maintenance task and the signaling message processing task have the same priority. Since the more session is allocated by the router the more session should be watched by the session maintenance task, the number of allocated sessions also has impact on this delay.

In order to simplify the characterization of the delay caused by the session maintenance task watching expired sessions, I have used the following equation for the approximation, instead of deriving the amount of this delay from the relation of the base signaling message processing time and the number of sessions:

\[ d_{\text{WATCH}}(r, m_i) = t_{m1} \cdot r + t_{m2} \cdot r^2, \]

where \( r \) is the number of sessions and the coefficients of the regression are \( t_{m1} \) and \( t_{m2} \).

Table 9 contains the coefficients that depend on the type of the signaling message \( (m_i) \) being processed.

<table>
<thead>
<tr>
<th></th>
<th>PATH</th>
<th>RESV</th>
<th>PATH</th>
<th>RESV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( t_{m1} ) [µs]</td>
<td>( t_{m2} ) [µs]</td>
<td>( t_{m1} ) [µs]</td>
<td>( t_{m2} ) [µs]</td>
</tr>
<tr>
<td>PATH</td>
<td>0.02311</td>
<td>0.0000563</td>
<td>0.03078</td>
<td>0.0000457</td>
</tr>
<tr>
<td>RESV</td>
<td>0.03078</td>
<td>0.0000457</td>
<td>0.0017</td>
<td>0.000033</td>
</tr>
<tr>
<td>PATH TEAR</td>
<td>0.0017</td>
<td>0.000033</td>
<td>0.001576</td>
<td>0.0000463</td>
</tr>
<tr>
<td>RESV TEAR</td>
<td>0.00017</td>
<td>0.000033</td>
<td>0.001576</td>
<td>0.0000463</td>
</tr>
</tbody>
</table>

In brief, the total length of all the interruptions caused by the session maintenance task watching expired sessions and also refreshing them is estimated with the following expression:

\[ d_{\text{SMO}}(t_{SO}, t_{MO}, r, t_S) = d_{\text{WATCH}}(r, m_i) + d_{\text{ROUT}}(r) \cdot f_{\text{ROUT}}(r). \]

Substituting the numerical values, it is:

\[ d_{\text{SMO}}(r) = t_{m1} \cdot r + t_{m2} \cdot r^2 + (53.83\,\mu\text{s} + 0.2852\,\mu\text{s} \cdot r) \cdot 0.00031 \cdot r \]
where \( r \) is the number of maintained sessions and \( l_{m1}, l_{m2} \) are coefficients that depend on the type of the signaling message that is processed in parallel with the session maintenance activity.

Knowing the \( d_{SMO} \) characteristic function I have determined the \( d_{SMI} \) function as well. The \( d_{SMI} \) function describes the delay that is caused by the session maintenance task handling incoming refresh messages. I have performed measurements and subtracting the corresponding results of the previous measurements I have determined the delay of the signaling message processing activity that is caused by the incoming signaling message handling. Figure 74 and Figure 75 display the result of the delay calculations.

The measurement results indicated that the incoming refresh messages have impact on this signaling message processing delay caused by the rival session maintenance task. Moreover, the results also have shown that the length of the delay depends on the total signaling message processing time. Thus, I have matched the formula that describes the delay when a higher priority task regularly interrupts the signaling message processing. I have assumed that the number of interruptions equals to the rate of the incoming signaling messages multiplied by the length of the total signaling message processing time. RSVP describes that each resource reservation session should be refreshed once in a half minute initially. However, in order to decrease the number of refresh messages, RSVP allows to increase this interval with each new refresh message. In addition to avoid the synchronization of the refresh messages, the routers use random timing for the refresh signaling message initiations. Still, despite the increase of the refresh period and the random timing for the refresh messages, the number of incoming refresh messages remains predictable. In my benchmarking measurements, to ease the calculations, I have rejected the refresh interval increase, thus throughout all the measurements all sessions were refreshed once in a half minute. Since RSVP requires to refresh not just the reservation sessions, but also the reservation paths, it doubles the number of refresh messages. In this case each session is refreshed with one refresh message in every 15 seconds. Based on these assumptions I have matched the following equation to the results:

\[
d_{SMI} (r,t_S) = t_S \cdot \frac{r}{15sec} \cdot d_{RIN},
\]

where \( r \) is the number of the maintained reservation sessions, \( t_S \) is the total signaling message processing time and \( d_{RIN} \) is the length of the interruption that an incoming refresh message generates.
Since the previous equation fits to the measurement results, I have determined that the session maintenance task handling incoming signaling messages has higher priority than the signaling message processing task. Furthermore, I have calculated that each incoming refresh signaling messages generates a 204 µs long interrupt in the signaling message processing.

I have summed up all the interruptions that the session maintenance activity of the ISI RSVP protocol causes in the signaling message processing task and got the following characteristic function:

\[
d_{SM}(r, t_S) = t_{m1} \cdot r + t_{m2} \cdot r^2 + (53.83 \mu s + 0.2852 \mu s \cdot r) \cdot 0.00031 \cdot r + t_S \cdot \frac{r}{15 \text{sec}} \cdot 204 \mu s,
\]

where \( r \) is the number of maintained reservation sessions; \( t_S \) is the total signaling message processing time; and finally, \( t_{m1} \) and \( t_{m2} \) are coefficients that depend on the base signaling message processing time.

5.2.5 The Data Forwarding Delay Caused by the Signaling Message Processing Task: \( d_{DS}(t_{D0}, t_{S0}, q_S, t_D) \)

I have determined the exact characteristics of the delay in the traffic forwarding time generated by the interruptions of the signaling message processing task. For this measurement I have sent data traffic and signaling flows through the tested router at the same time, changing the rate of the signaling messages during the tests. The exact benchmarking methodology is the following:

I have constructed a traffic test flow from equally sized data packets using unpredictable long breaks between them. I have also generated a signaling test flow that consisted of the same type of signaling messages or signaling message pairs depending on the type of the message, but here the rate of the messages was constant. I have sent the two flows through the router at the same time, but using different interfaces of the router as the data traffic and signaling message entry points. I have measured the traffic forwarding time on all the forwarded data packets. Finally, I have calculated the investigated delay by reducing the measured results by the corresponding base traffic forwarding time. I have repeated the measurements using signaling flows constructed from other types of signaling messages and using different signaling message intensities.

Measuring the Boomerang and ISI RSVP protocol implementations I have got similar measurement results presented in Figure 76.
Figure 76 - Data Traffic Forwarding Delay vs. Signaling Message Intensity

From the measurement results I have concluded that the delay caused by the interruptions of the signaling message processing task does not depend on the type of the signaling messages. Previously, in the case of both protocols, I have shown that the traffic forwarding task has higher priority than the signaling message processing task. However, this result indicates that the data traffic forwarding time depends on the activity of the signaling message processing task. The reason is that a part of the signaling message processing task, which transfers the packets of the signaling messages from the network card to the main memory, is a kernel routine running at higher priority than the traffic forwarding task. This activity interrupts the data traffic forwarding task in the case of both protocols. I have matched the formula describing this interruption case to the results, and got the following expression:

\[ d_{DS} (q_S, t_D) = t_D \cdot q_S \cdot d_S, \]

where \( t_D \) is the total data forwarding time, \( q_S \) is the rate of the signaling messages expressing the signaling message intensity and \( d_S \) is the length of the interruption that one signaling message takes when the operating system transfers it from the network card to the main memory.

From the computations I have got 13.6 \( \mu s \) as the length of the interruptions in the case of both resource reservation protocols. This way the delay in the traffic forwarding task caused by the interruptions of the signaling message processing task is characterized as follows:

\[ d_{DS} (q_S, t_D) = t_D \cdot q_S \cdot 13.6 \mu s, \]

where \( t_D \) is the total data forwarding time and \( q_S \) is the rate of the signaling messages.

5.2.6 The Data Forwarding Delay Caused by the Session Maintenance Task: \( d_{DM}(t_{DO}, t_{MO}, r, t_D) \)

At last, I have characterized the interruptions of the data forwarding task generated by the session maintenance activity. Similarly to the investigation of the signaling message processing delay caused by the interruptions of session maintenance task, here I have also differentiated the instant response and non-instant response protocols. In the case of non-instant response protocols I have separated the benchmark into two measurements. First, I have investigated the delay caused by the session expiration watch and refresh message generation activity of the session maintenance task. Second, I have examined how the session maintenance affects the data traffic forwarding by handling incoming refresh messages. The sum of the two delay
functions are the delay that is caused by the session maintenance task in the data traffic forwarding. In the case of instant response protocols I have measured the effect of the incoming and outgoing refresh message managements at the same time. The exact measurement description is the following:

*In the case of non instant response protocols, I have constructed two measurements. In the first measurement I have set up a number of resource reservation sessions in the tested router, but the signaling end-points have not sent refresh signaling messages. I have composed a traffic test flow from constant sized data packets using unpredictable long breaks between them. I have sent the test traffic through the router measuring the data traffic forwarding time. The delay caused by the refresh message generation of the session maintenance activity is calculated by reducing the results of the measurements by the corresponding base traffic forwarding time. I have repeated this measurement for various numbers of allocated sessions. In the second measurement I have also set up a number of allocated sessions in the tested router, but this time the signaling endpoints maintained the sessions by sending refresh signaling messages. The rest of the measurement is the same as in the previous case, except the calculation of the investigated delay. In this case, from the measurement results I have subtracted the measured results of the first measurement and thus I have got the direct influence of the incoming refresh messages. Similarly to the first measurement, I have repeated this measurement using different amounts of session allocations in the tested router. The delay caused by the session maintenance activity in the data traffic forwarding is the sum of the delays determined in the case of incoming and outgoing refresh message measurements.*

*In the case of instant response protocols I have allocated a number of resource reservation sessions in the tested router. I have constructed a traffic test flow the same way as the previous measurements and sent it through the router. In order to get the investigated delay I have subtracted the corresponding base traffic forwarding time from the measurement results. I have repeated the measurement using various numbers of allocated sessions in the router.*

In the case of the Boomerang protocol, which is an instant response protocol, the measurement results investigating the router maintaining up to 10000 sessions, show that the session maintenance activity has only a slight impact on the traffic forwarding task. The amount of the delay is so small that it is impossible to say the exact type of the dependence. Based on the previous measurements I have presumed that the traffic forwarding task has higher priority than the session maintenance task, but the incoming refresh messages still interrupt the data traffic forwarding, like the signaling messages processing task interrupts the data traffic forwarding activity. In this case the rate of the signaling messages depends on the number of allocated sessions. Since the protocol describes that each session should be refreshed once within every 30 seconds, the signaling message intensity is the number of sessions divided by 30 seconds. Due to the similarity, I have used the same formula and interrupt length parameter as in the case of the delay that is caused by the interruptions of the signaling message processing task:

\[
 d_{DM}(r, t_D) = t_D \cdot \frac{13.6\mu s}{30}\text{sec},
\]

where \( r \) is the number of allocated sessions.

In the case of the ISI RSVP protocol, I have performed two separate benchmarking measurements, since the RSVP protocol is a non-instant response protocol. Similarly
to the measurements where I have examined the delay of the signaling message processing task caused by the session maintenance activity, here I have used also two functions to characterize the interruptions of the data traffic forwarding task:

$$d_{DM} \left(t_{D0}, t_{M0}, r, t_D\right) = d_{DMI} \left(t_{D0}, t_{M0}, r, t_D\right) + d_{DMO} \left(t_{D0}, t_{M0}, r, t_D\right),$$

where the $d_{DMI}$ function describes the length of the interruptions caused by the incoming refresh messages and the $d_{DMO}$ function gives the delay rooted in the generation of the outgoing refresh messages. The other parameters are the base traffic forwarding delay, denoted by $t_{D0}$; $t_{M0}$ that is the length of one session maintenance activity; $r$ is the number of allocated sessions and $t_D$ is the total length of the traffic forwarding activity.

The first measurement investigating the delay caused by the session maintenance activity watching the expired sessions and sending refresh messages to renew them in neighboring routers resulted that the data traffic forwarding is not influenced by this kind of session maintenance activity:

$$d_{DMO} \left(t_{D0}, t_{M0}, r, t_D\right) = 0.$$

The second benchmarking measurement that analyze the interruptions generated by the session maintenance task handling incoming refresh messages also resulted that the data traffic forwarding task of the tested router maintaining up to 1500 resource reservation sessions is not affected by the session maintenance activity perceptibly. However, in the previous tests I have identified that in fact, the incoming signaling messages cause interruptions in the data traffic forwarding task. This delay in the case of 1500 sessions is on a nanosecond magnitude, therefore it is hardly any observable using my measurement module. This way, similarly to the analysis of the Boomerang protocol, I have assumed that the data traffic forwarding task has higher priority than the session maintenance task, but the incoming refresh messages interrupts the data forwarding for a really short time. The number of interruption can be calculated from the refresh timing information that the sessions define. In the case of my measurement setup there is one reservation and one path refresh messages in every 30 seconds, so the rate of the signaling messages is the number of sessions divided by 15 seconds. The rest is analogue to the case of the Boomerang protocol. The expression describing the effect of the interruptions is the following:

$$d_{DMO} \left(r, t_D\right) = t_D \cdot \frac{r}{30sec} \cdot 13.6 \mu s,$$

where $r$ denotes the number of allocated sessions.

After all, the influence of the session maintenance task over the data forwarding activity, in the case of the ISI RSVP protocol is characterized with the following expression:

$$d_{DM} \left(r, t_D\right) = t_D \cdot \frac{r}{30sec} \cdot 13.6 \mu s,$$

where the number of sessions is denoted by $r$.

### 5.2.7 Benchmarking Measurement Summary

Describing all the performance characteristic functions I have determined the exact performance characteristics equations for the signaling message processing time and
data traffic forwarding time metrics in the case of the Boomerang and ISI RSVP protocols. In Table 10 and Table 11 I have summarized these performance characteristic functions. In the case of the base traffic forwarding time \( t_{D0} \), there are two different performance characteristic functions depending on the size of the packets. In this table I have presented that case when the packets are larger than 256 bytes.

### Table 10 - Performance Characteristic Functions for the Boomerang Protocol

| \( \text{t}_{S0}(r,m) \) | \( 42 \mu s + 0.1374 \mu s \cdot r^{0.3966} \) |
| \( \text{d}_{SD}(t_{S0},t_{D0},q_{T},t_{S}) \) | \( t_{S} \cdot q_{T} \cdot (t_{D0} + 27.8 \mu s) \) |
| \( \text{d}_{SM}(t_{S0},t_{M0},r,t_{S}) \) | \( t_{S} \cdot \frac{r}{30 \text{sec}} \cdot 14.2 \mu s \) |
| \( \text{t}_{D0}(r,s) \) | \( 14.6 \mu s + 0.692 \mu s \cdot r^{0.2283} \) |
| \( \text{d}_{DS}(t_{D0},t_{S0},q_{S},t_{D}) \) | \( t_{D} \cdot q_{S} \cdot 13.6 \mu s \) |
| \( \text{d}_{DM}(t_{D0},t_{M0},r,t_{D}) \) | \( t_{D} \cdot \frac{r}{30 \text{sec}} \cdot 13.6 \mu s \) |

### Table 11 - Performance Characteristic Functions for the ISI RSVP Protocol

| \( \text{t}_{S0}(r,m) \) | \( t_{r0} + t_{r1} \cdot r + t_{r2} \cdot r^2 \) |
| \( \text{d}_{SD}(t_{S0},t_{D0},q_{T},t_{S}) \) | \( t_{S} \cdot q_{T} \cdot (t_{D0} + 40.3 \mu s) \) |
| \( \text{d}_{SM}(t_{D0},t_{M0},r,t_{S}) \) | \( t_{m1} \cdot r + t_{m2} \cdot r^2 + \left( 53.83 \mu s + 0.2852 \mu s \cdot r \right) \cdot 0.00031 \cdot r + t_{S} \cdot \frac{r}{15 \text{sec}} \cdot 204 \mu s \) |
| \( \text{t}_{D0}(r,s) \) | \( 14.8 \mu s + 0.0005 \mu s \cdot r \) |
| \( \text{d}_{DS}(t_{D0},t_{S0},q_{S},t_{D}) \) | \( t_{D} \cdot q_{S} \cdot 13.6 \mu s \) |
| \( \text{d}_{DM}(t_{D0},t_{M0},r,t_{D}) \) | \( t_{D} \cdot \frac{r}{15 \text{sec}} \cdot 13.6 \mu s \) |

Finally, I have calculated the whole performance characteristic equations for the investigated resource reservation capable routers by aggregating the previous performance characteristic functions into a single equation. Table 12 and Table 13 show these performance characteristic equations, where the appropriate coefficients can be found in previous tables.

### Table 12 - Performance Characteristic Equations for the Boomerang Protocol

| \( \text{t}_{S} \) metric | \( \frac{42 \mu s + 0.1374 \mu s \cdot r^{0.3966}}{1 - \frac{14.6 \mu s}{30 \text{sec}} - q_{T} \cdot (42.4 \mu s + 0.692 \mu s \cdot r^{0.2283})} \) |
| \( \text{t}_{D} \) metric | \( \frac{14.6 \mu s + 0.692 \mu s \cdot r^{0.2283}}{1 - \left( q_{S} + \frac{r}{30 \text{sec}} \right) \cdot 13.6 \mu s} \) |
Table 13 - Performance Characteristic Equations for the ISI RSVP Protocol

<table>
<thead>
<tr>
<th>ts metric</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( t_{s} = t_{i0} + t_{i1} \cdot r + t_{i2} \cdot r^2 + t_{m1} \cdot r + t_{m2} \cdot r^2 + \frac{0.0166873 \mu s \cdot r + 0.000088412 \mu s \cdot r^2}{1 - \frac{r}{15 \text{sec}}} \cdot 204 \mu s - q_r \cdot (55.1 \mu s + 0.0005 \mu s \cdot r) )</td>
</tr>
<tr>
<td>t0 metric</td>
<td>Equation</td>
</tr>
<tr>
<td></td>
<td>( t_{0} = 14.8 \mu s + 0.0005 \mu s \cdot r )</td>
</tr>
</tbody>
</table>

5.3 Validation of the Results

In order to validate the performance characteristic equations, I have performed measurements where I have checked the results of the measurements against the values that I have calculated using the performance characteristic equations. Table 14 shows five test cases that I have tested for both resource reservation protocol implementations:

Table 14 - Validation Checkpoints for the Performance Characteristic Equations

<table>
<thead>
<tr>
<th>Test</th>
<th>Number of Sessions</th>
<th>Data Packet Size [byte]</th>
<th>Data Traffic Rate [packets/s]</th>
<th>Signaling Message Intensity [msg/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>512</td>
<td>2000</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>1024</td>
<td>3000</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1024</td>
<td>2000</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>1500</td>
<td>512</td>
<td>2000</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>1500</td>
<td>1024</td>
<td>3000</td>
<td>50</td>
</tr>
</tbody>
</table>

In the case of the Boomerang resource reservation protocol Table 15 presents both the results of the measurements and the calculations, plus the difference of these two values:

Table 15 - Verification of the Performance Characteristic Equations Describing the Boomerang Resource Reservation Protocol

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17.54 \mu s</td>
<td>17.46 \mu s</td>
<td>0.44</td>
<td>43.72 \mu s</td>
<td>43.77 \mu s</td>
<td>-0.12</td>
</tr>
<tr>
<td>2</td>
<td>17.52 \mu s</td>
<td>17.46 \mu s</td>
<td>0.32</td>
<td>43.85 \mu s</td>
<td>43.87 \mu s</td>
<td>-0.05</td>
</tr>
<tr>
<td>3</td>
<td>17.86 \mu s</td>
<td>17.95 \mu s</td>
<td>-0.48</td>
<td>44.28 \mu s</td>
<td>44.34 \mu s</td>
<td>-0.13</td>
</tr>
<tr>
<td>4</td>
<td>18.21 \mu s</td>
<td>18.28 \mu s</td>
<td>-0.41</td>
<td>44.67 \mu s</td>
<td>44.75 \mu s</td>
<td>-0.18</td>
</tr>
<tr>
<td>5</td>
<td>18.30 \mu s</td>
<td>18.28 \mu s</td>
<td>0.09</td>
<td>44.72 \mu s</td>
<td>44.90 \mu s</td>
<td>-0.40</td>
</tr>
</tbody>
</table>

In the case of the ISI RSVP resource reservation protocol implementation, in Table 16I have displayed the measurement results and the calculations regarding the data traffic forwarding time, while
Table 17 presents the results of the measurements and calculations obtained on the Path and Resv signaling messages:

Table 16 - Verification of the Performance Characteristic Equations Describing the ISI RSVP Resource Reservation Protocol

<table>
<thead>
<tr>
<th>Test</th>
<th>Measured Traffic Fwd. Time</th>
<th>Calculated Traffic Fwd. Time</th>
<th>Diff. [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.83 µs</td>
<td>15.05 µs</td>
<td>-1.46</td>
</tr>
<tr>
<td>2</td>
<td>15.37 µs</td>
<td>15.05 µs</td>
<td>2.12</td>
</tr>
<tr>
<td>3</td>
<td>15.41 µs</td>
<td>15.30 µs</td>
<td>0.72</td>
</tr>
<tr>
<td>4</td>
<td>15.77 µs</td>
<td>15.55 µs</td>
<td>1.41</td>
</tr>
<tr>
<td>5</td>
<td>15.48 µs</td>
<td>15.55 µs</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 17 - Verification of the Performance Characteristic Equations Describing the ISI RSVP Resource Reservation Protocol (continued)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>462.82 µs</td>
<td>466.94 µs</td>
<td>-0.88</td>
<td>1096.03 µs</td>
<td>1096.73 µs</td>
<td>-0.00</td>
</tr>
<tr>
<td>2</td>
<td>490.52 µs</td>
<td>498.19 µs</td>
<td>-1.54</td>
<td>1150.15 µs</td>
<td>1170.12 µs</td>
<td>1.70</td>
</tr>
<tr>
<td>3</td>
<td>737.91 µs</td>
<td>722.01 µs</td>
<td>2.20</td>
<td>1356.20 µs</td>
<td>1358.86 µs</td>
<td>-0.19</td>
</tr>
<tr>
<td>4</td>
<td>1088.40 µs</td>
<td>1074.54 µs</td>
<td>1.29</td>
<td>1710.26 µs</td>
<td>1712.46 µs</td>
<td>-0.13</td>
</tr>
<tr>
<td>5</td>
<td>1139.05 µs</td>
<td>1148.44 µs</td>
<td>-0.82</td>
<td>1834.62 µs</td>
<td>1830.24 µs</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Since the difference between the measured and the calculated performance metrics are lower than 3 percent for all the test cases, which is an acceptable difference, thus I have proven that the performance characteristic equations determined via the proposed performance analysis framework are close to reality. Based on the near identical measurement and calculation results I have also shown that the resource reservation capable router model describes single processor Linux routers correctly. This way I have validated my performance analysis framework proposal.

5.4 Future Extensions to the Performance Analysis Framework

The performance analysis framework that I have proposed is based on a router model that describes single processor Linux routers running single threaded resource reservation protocols. However, extending the model it would be also capable of portraying even more kinds of router products.

Measurements results [J1] already have proven that in the case of Cisco 7200 routers there is a coupling between the performance of the traffic forwarder block and the performance of the signaling message processor. However, due to the lack of a thorough performance analysis of that Cisco router, it is not yet proven that the specific router product is measurable using my performance analysis framework.
Even through, since most of the recent router products have small operating systems running on the top of routing architecture, I suppose that the resource reservation protocol, just like the management interface of the router, is an application of this operating system. Consequently, I assume that even if the routing and forwarding is performed by special hardware elements, the resource reservation protocol runs on the single central processor competing with other tasks. In this case the single processor Linux router model can be applied to other router products using software implementation of a resource reservation protocol. Unfortunately, most of the network device vendors do not allow reverse engineering on their products, and therefore implementation details of the resource reservation protocol and the processor power scheduling remain undiscovered.
6 Summary of the Dissertation

In my dissertation I have proposed an adaptive teleservice broker system and using simulation analyses I have proven that this system negotiates higher quality teleservices than earlier approaches, while the required calculation complexity of the algorithms is still satisfactory. Thus, the introduction of the adaptive teleservice broker system would improve the network utilization and produce extra revenue for network operators and service providers. Even the participants of the teleservices would benefit from the teleservice broker system by acquiring higher quality adaptive services that adapt to the fluctuating network resource conditions without user interactions.

In this dissertation work I have also proposed a resource reservation capable router performance analysis framework. This kind of router characterization gives a tool to networking device designers to benchmark their resource reservation protocol implementations running on Linux routers. Although the proposed performance analysis framework is limited to single processor, single threaded and priority scheduler based routers, some experimental measurements have already shown that lots of commercial routers have similar features and thus my performance analysis framework is applicable to them as well. Using such tool, network operators can be aware of the resource reservation handling related performance of the routers and thus they are able to design, scale and optimize their networks to have better performance delivering quality of service. The performance characteristic equations, due to their simplicity, can be really efficient in large network simulations, where the simulations of all the single processor, single threaded and priority scheduling based resource reservation capable routes can be substituted with their characteristic equations. Moreover, since the elements of the performance characteristic equations refer to the building blocks and their relations, these equations suggest the performance modifications when the internals of a tested router change, providing guidelines to the router code developers.

Most of researches in this dissertation work are already published on several conferences and appeared in numerous printed publications. Despite that most of the publications are prepared in cooperation with my research fellows, all the research work mentioned in this dissertation is my own result.

Acknowledgements

I would like to thank to all the people who consulted and guided me through the research of the dissertation work, especially to Dr. István Cselényi, Dr. József Bíró, Dr. Tamás Henk, Dr. Róbert Szabó and Krisztián Németh. I would also like to thank the support of the Telia Research AB, Sweden and the High Speed Networks Laboratory of the Budapest University of Technology and Economics.
References


Publication of the New Results

[B] Papers in Edited Books


[J] Journal Papers


[C] Conference Papers


[O] Other Publications


