An Analysis of QoS Provisioning for High Definition Video Distribution in Heterogeneous Network

Robert R. Chodorek
Department of Telecommunications
The AGH University of Science
and Technology
Kraków, Poland
e-mail: chodorek@kt.agh.edu.pl

Agnieszka Chodorek
Department of Telecommunications, Photonics
and Nanomaterials
Kielce University of Technology
Kielce, Poland
e-mail: a.chodorek@tu.kielce.pl

Abstract— **In this paper we propose a novel method of dynamic RSVP's reservations, based on knowledge about the sending stream. The proposed solution is able to assure better link utilization than classis RSVP reservations and offer no worse QoS for HD video transmission than the typical RSVP reservations.**

Keywords- network emulation, HD video streaming, QoS, heterogeneous IP network, RSVP

I. INTRODUCTION

Many modern Internet services (e.g. VoIP, IPTV) require real-time multimedia transmission [1]. A commonly used best effort service gives high link utilization, although it cannot offer suitable QoS parameters for real-time video distribution, especially when high definition (HD) video is transmitted. Usage of Resource ReSerVation Protocol (RSVP) allows a distribution system to achieve satisfactory QoS [2][3] at the cost of lower link utilization than is offered by the best effort. Therefore, network resources do not have optimal utilization when full QoS guarantees are assured. This results from the necessity of reserving resources for the variable bit rate traffic with characteristics that are difficult to define simply.

In this paper we propose a novel method of dynamic RSVP reservations which deals with the problem of a trade-off between link utilization and quality of real-time service. This problem is especially noticeable in the case of streaming HD video, because of its large bit rates, comparable with typical bit rates of aggregated traffic. The proposed extension to the RSVP protocol allows the RSVP to make elastic adjustment of reservations to the current demand for network resources. The proposed method dynamically changes RSVP reservations on the basis of knowledge about transmitted traffic. This knowledge is taken from the sending buffer of the video streaming application. Such a buffer functions in typical implementations of streaming applications, and so the proposed solution doesn't interfere with the standard streaming mechanism.

The paper is organized as follows. Section II is devoted to proposed extensions of the RSVP protocol. Section III presents the emulation environment, used for the investigations. We pay a special attention to our original extensions to the emulator. Section IV describes experiments, while Section V shows obtained results. Section VI concludes the paper.

The work presented in this paper was supported by the Ministry of Science and Higher Education of the Republic of Poland, under the project No. N N517 4388 33.

II. EXTENSIONS OF THE RSVP PROTOCOL

The RSVP is a signaling protocol intended for the assurance of required quality of service (QoS). The RSVP session starts before the beginning of the multimedia transmission and ends after the transmission of the multimedia stream is finished.

Typically, the RSVP session consists of the following events. At the first, a multimedia application passes parameters of the stream to the RSVP protocol. The RSVP sends parameters to a receiver (or receivers) using the PATH message. Routers, which forwarding the PATH message, store path state and, if necessary, modify information about bottleneck fragments of the path conveyed in the message. The receiver, which receives the PATH message, makes autonomous decisions about reservations and sends reservation requests (the RESV message), which create and maintain reservations for further multimedia transmission.

Each node which accepts a reservation sends the RESV to the next node, upstream, towards the sender. As a result, the RESV follows exactly the same reverse path of the PATH message. Because the PATH is naturally transmitted via routers, which will be intermediate nodes for further multimedia transmission, each router along the data path makes reservations of resources to assure required QoS guarantees.

If all nodes will accept the reservation, the RESV message is delivered to the sender node. If any node will not be able to accept the reservation, the reservation is rejected and error message is sent to the receiver.

The state of reservation is soft, so the receiver periodically refreshes the reservation sending the RESV messages. After the transmission is finished, the session can be explicitly torn down by exchange of the PATHTEAR and the RESVTEAR messages or implicitly torn down by lack of RESV messages.

The RSVP reservations are static in nature and typically they last for the whole session. There is no simple mechanism to enable reservation parameters to change dynamically in response to changed network resources. It is especially troubling in the case of HD video transmission, when a single variable bit rate (VBR) stream usually occupies an important part of the available resources.

In this paper we propose extensions to the RSVP protocol, which enable reservations to be changed dynamically in the event the network resource requirements change for the transmitted multimedia stream. The proposed extension, in many cases, allows for the release of part of the resources and enables their utilization for different transmissions. In practice it is possible to use released resources for transmission of elastic traffic (e.g. the traffic observed during bulk data transmissions carried out using the TCP or other reliable transport protocol).

The proposed extensions use an additional message, named CHANGE. CHANGE messages are periodically sent by the sender and they are transmitted downstream towards a receiver (or receivers), using exactly the same path which the multimedia stream uses. They convey information about requirements for network resources which will be expected soon. Information is acquired by the analysis of the sender buffer of application which sends the multimedia stream.

Based on information conveyed in the CHANGE message, the intermediate nodes can release resources, which won't be used by the multimedia traffic in the next period of time. Thus, these resources become available for other transmissions, which should improve utilization of the network.

CHANGE messages carry k values of bit rates R_i, $i = 1, 2, 3, k$, of transmitted multimedia stream, averaged over a given period of time. Intermediate nodes analyze these bit rates and make decisions to instantaneously change the value of the reservation parameters. The simplest decision algorithm calculates the new reservation R_{new} according to equation:

$$R_{new} = \min\{R_{RESV}, \max(R_1, R_2, \ldots, R_k)\} \quad (1)$$

where R_{RESV} - a throughput value carried in the RESV message.

The C++ code of the decision algorithm is shown in Fig. 1. The variable `R_CURR` stores the current value of the reservation and the `R_NEW` stores value of the R_{new}, estimated from the Equation 1. The `modify_resv` function modifies the instantaneous settings of the reservation, established for a given data stream, in a given router. Note that in the case of lack of periodical CHANGE messages, R_{curr} will return to the R_{RESV}.

Other algorithms, e.g. based on traffic prediction, can also be used. These algorithms will be able to estimate future values of required throughput in a more sophisticated way.

Released network resources can be used by protocols which have a natural ability to occupy available bandwidth. One such protocol is the TCP transport protocol. The TCP uses the additive increase/multiplicative-decrease (AIMD) mechanism, which gives both congestion control and the ability to utilize non-used capacity of bottleneck links.

```
if (R_NEW < R_CURR) {
        if ((R_CURR - R_NEW) > B1 * R_CURR) {
                R_CURR = R_NEW;
                modify_resv(R_CURR); }
} else if (R_NEW > R_CURR) {
        R_CURR = R_NEW;
        modify_resv(R_CURR); }
```

Figure 1. Code of the decision algorithm.

III. THE EMULATOR

In our investigations, the Berkeley's ns-2 simulator working as an emulation facility was used [4]. The emulation mode enables the co-operation of the simulator with a real network. This mode allows the ns-2 to capture packets arriving from the real network and, after processing them in a simulated network environment, to inject them into the real network.

The ns-2 is an event-driven simulator, while the co-operation of a real network requires a time-driven system - i.e. all events in the emulated part of network must be run in real-time. Thus, the ns-2 emulation facility replaces the default system scheduler with the new real-time scheduler, which is able to preserve real-time conditions of the emulated part.

The emulator, used in our research, was supplemented by extensions developed at the University of Magdeburg (Germany) [5][6]. Supplementary software includes extensions for the real-time scheduler module and the modification of modules enabling co-operation with a real network (network objects module and tap agents module).

In spite of the usage of extensions mentioned above, the emulator wasn't able to carry out real-time HD video properly. Such a transmission considerably overloaded the emulation system, so further modifications of the emulator were needed.

To enable emulation of real-time HDTV video streams, we supplemented the emulator with our original extensions for real-time scheduler module and interface module.

Extensions for the real-time scheduler include the co-operation of the emulator with a 64-bit processor and with the Linux core able to process 64-bit instructions (the emulator was optimized for any 64-bit version of the Linux operating system). It allows the creation of fully 64-bit code from the real-time scheduler - the most neuralgic element of the emulator from the real-time point of view. The real-time scheduler is the main limitation for the speed and precision of operations in the real-time.

Extensions developed for interface service allow the emulator to cooperate with both directly connected computers and other systems connected via a more complicated heterogeneous IP network. These extensions allow the test of connections between applications run at end-systems located in any fragment of the Internet. They also allow the connection of various emulation systems, which emulated different fragments of the network and which are run on remote computers. Finally, developed extensions enable processing of both unicast and multicast IP datagrams.

As the result, the emulator can be connected both to computers from its local network and to computers from the global Internet. Moreover, the emulated topology can be divided into n sub-emulators, each emulating its own fragment of the network, what allows both for distributed emulation and to include real infrastructure into the experimental system.

Figure 2 depicts an example of the emulation system, equipped with two interfaces $i1$ and $i2$. These interfaces are connected to local subnetworks, which are real (live) IP networks. Local subnetworks can be both separate networks or subnetworks of the global Internet network.

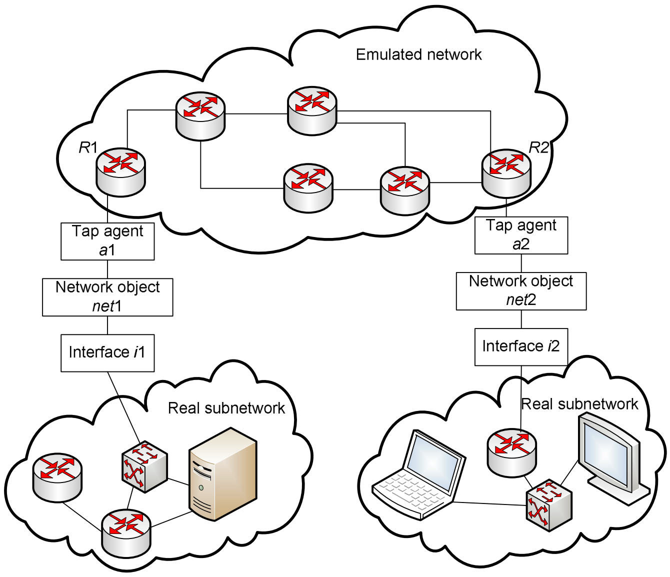

Figure 2. The emulation system.

Subnetworks include elements of tested infrastructure and crucial elements of the HD video distribution system. Nodes R1 and R2 of the emulated network are edge routers, located at the border of the emulated and real network. Through R1 and R2 nodes, traffic enters and exits the emulated network.

Tap agent modules (a1 and a2) are connected to R1 and R2 nodes. Tap agent modules a1 and a2 are program interfaces, which carry out the preprocessing of packets injected from the real network to the emulator environment. One important function of tap agents is to translate the format of injected packet to the simplified format used by the ns-2 protocol. Tap agents (a1 and a2) communicate with the real network via network modules (net 1 and net2 in Figure 2).

Both tap agent modules and network modules, used in our research, are extended versions of the original University of Magdeburg's modules. They allow dynamic (instead of original - static) connection of the emulator with external (real, live) networks on the level of both the network layer and the link layer of the OSI model (originally only the link layer connections were considered). Our extension enables nearly full cooperation with the external network working on the IP platform (including simplified routing).

Developed extensions use a new structure `route_pkt_`, defined for interfaces serving input and output traffic. The structure has a format similar to typical routing tables, so the emulator is seen by external networks as a single router. Configuration of this "virtual router" is analogous to the definition of the static routing in typical routers. During unicast transmission ARP (IPv4) or ND (IPv6) protocols are used to discover MAC address.

IV. EXPERIMENTS

A. *HDTV Video Sequences*

In this paper High Definition Television (HDTV) video sequences were used for the experiments. Video content is collected in a library of HDTV video, publicly available at The Video Quality Experts Group (VQEG) [7] site, at URL ftp://vqeg.its.b. These video sequences are owned by NTIA/ITS, an agency of the U.S. Federal Government. They were created under Project Number 3141012-300, Video Quality Research, in 2008.ldrdoc.gov/HDTV/NTIA_source/.

The collection consists of 8 clips, each last for 19 seconds. Each clip includes full high definition 1920 x 1080p native video, captured at 30 frames per second. The material is characterized by large and very large dynamics of the video content, large amount of detail and (with the exception of WestWindEasy clip) sharp, sudden scene changes. The content can be described as follows:

- Aspen – forest landscape, leaves of trees in the wind.
- RedKayak – whitewater kayaking.
- WestWindEasy – combined picture: the scrolling text of the "Ode to the West Wind" by P. B. Shelley at the left and leaves of grass in the wind at the right.
- RushFieldCuts – a rush of spectators at the end of the game onto the football pitch.
- SnowMnt – snow-capped mountains.
- SpeedBag – speed bag training,
- TouchdownPass – a "passing" touchdown in an American football game.
- ControlledBurn – a controlled burn of a house that had been used as a methamphetamine lab.

The above sequences have been encoded using the H.264 codec into a Variable Bit Rate (VBR) data stream. The support of the sample bit depth precision was set to 8 bits per sample. Target bit rate of the VBR stream was set to 20 Mbps. The instantaneous value of the bit rate of exemplary VBR traffic is depicted in Fig. 3. Statistical properties of the video traffic, generated using test video sequences, are shown in Table 1.

The video traffic was observed at the input of the emulator (see Section C for details). Mean bit rate exceeded 20 Mbps (target bit rate), what can only partially be explained by packet overheads (IP, UDP, RTP and MPEG TS headers).

TABLE I. STATISTICAL PROPERTIES OF HDTV H.264 VIDEO TRAFFIC.

Clip number	Parameter				
	Mean	*Maximum*	*Variance*	*Kurtosis*	*Skewness*
0	$2.64*10^7$	$5.89*10^7$	$5.06*10^{13}$	1.619	0.9397
1	$2.58*10^7$	$4.77*10^7$	$4.08*10^{13}$	0.306	0.8042
2	$2.65*10^7$	$4.54*10^7$	$6.31*10^{13}$	-0.552	0.6164
3	$2.52*10^7$	$3.32*10^7$	$1.57*10^{13}$	4.077	-0.4221
4	$2.38*10^7$	$5.33*10^7$	$6.46*10^{13}$	1.382	0.8535
5	$2.65*10^7$	$4.21*10^7$	$2.38*10^{13}$	0.8735	-0.2022
6	$2.50*10^7$	$4.60*10^7$	$5.14*10^{13}$	0.389	0.8586
7	$2.66*10^7$	$3.55*10^7$	$8.49*10^{12}$	11.06	-0.8053

a. Numbers of video clips: 0 – Aspen, 1 – RedKayak, 2 – WestWindEasy, 3 – RushFieldCuts, 4 – SnowMnt, 5 – SpeedBag, 6 – TouchdownPass, 7 - ControlledBurn.

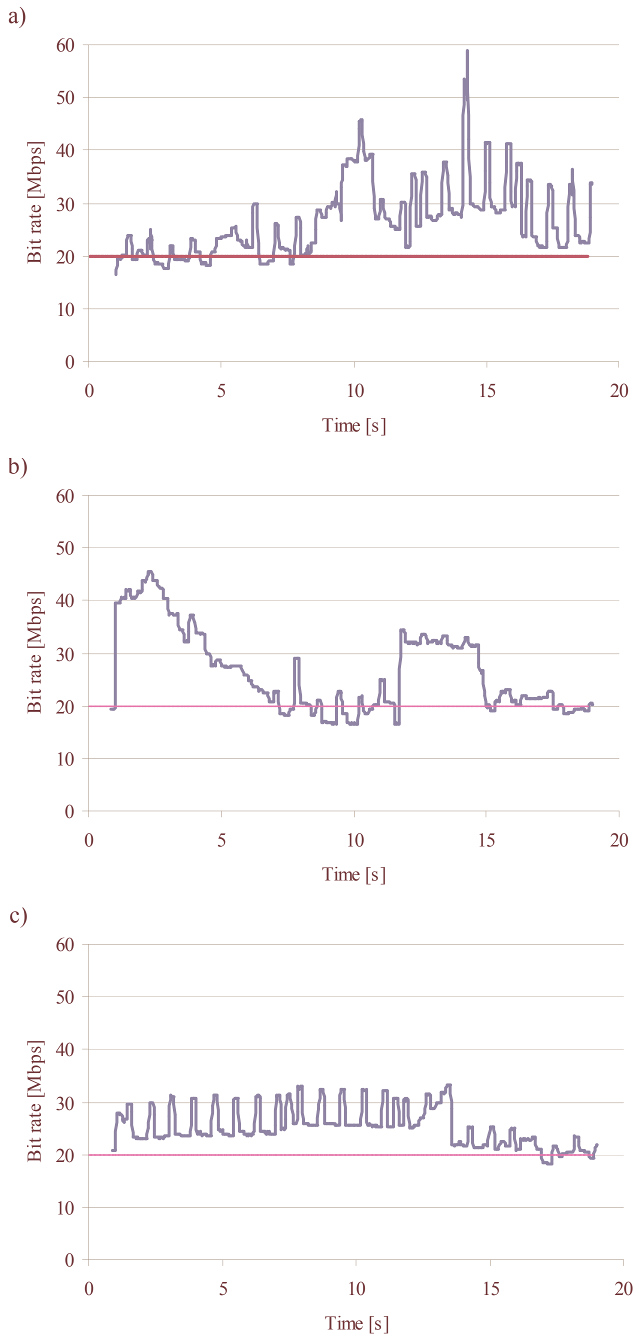

Figure 3. The instantaneous value of bit rate of VBR traffic: a) Aspen, b) WestWindEasy, c) RushFieldCuts.

An approximate estimation of packet overheads shows that only 20% of the violation of target bit rate is caused by overheads, while the remaining 80% is probably caused by the properties of the H.264 codec that was used. Scene changes and the large dynamics of video content cause large fluctuations of bit rate. Maximum bit rate of analyzed video traffic can be close to 60 Mbps (Fig. 3a), i.e. is 3 times larger than target bit rate.

B. The Emulation Environment.

We tested the proposed method using emulation system depicted in the Figure 4. The system consists of a streaming server (built with usage of the VLC tool), the network emulator and video receiver(s) (also built using the VLC tool). For the network emulator, the Berkeley's ns-2 simulator (working in emulation mode) was used.

The video server (PBZmatrix) was built on the Intel® Server Board S5000PSL platform with a dual Gigabit Ethernet card. The video server was equipped with two Intel® Xeon® E5410 (2.33GHz) processors, 4 GB of RAM memory and two hard disks of 1 TB and 2 TB. The network emulator (PBZ) was built on the basis of the same server platform as the PBZmatrix (Intel® Server Board S5000PSL). The PBZ was equipped with two Intel® Xeon® E5420 (2.50GHz) processors, 16 GB of RAM memory and two hard disks (smaller then PBZmatrix). As end systems (PCclient) highly efficient PC computers were used. The PCclient(s) were equipped with Intel® multicore processors and a Gigabit Ethernet interface (eth0).

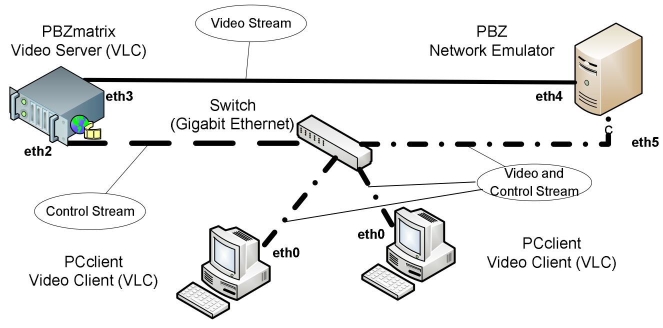

Figure 4. Block diagram of the emulation system – physical topology.

In order to obtain the not-distorted HD video stream in real-time, the video stream was sent directly from the video server to the network emulator through the dedicated link (the link from eth3 to eth4, in Figure 4). The control stream (signaling data) for this transmission is held using additional interfaces of both systems (PBZmatrix – interface eth2, PBZ – interface eth5), connected through the gigabit switch. End systems (PCclients) are also attached to this switch. The network emulator (PBZ) also sent the video stream to end systems through the interface eth5 (Figure 4).

Logical topologies of the emulated network are shown in Figure 5. The test network consists of four (Figure 5a) or five (Figure 5b) routers (from *R*1 to *R*5), a Gigabit Ethernet switch and a set of senders and receivers. In all experiments the throughput of links was large enough to ensure live video transmission – 65 Mbps between routers, 65 Mbps between *R*4 and switch, 1 Gbps between server and *R*1, and 1 Gbps between each video receiver and switch. The video sender sent the HD video stream to two video receivers. The video clips described in the previous section were used as video content.

TABLE II. PROPAGATION DELAYS.

Variant number	Level of aggressiveness	Propagation delay of link between:			
		R1 - R2	***R2 - R3***	***R3 - R4***	***R1 - R5***
1	Low	10 ms	10 ms	180 ms	-
2	Medium	3 ms	3 ms	194 ms	-
3	High	1 ms	3 ms	196 ms	1 ms

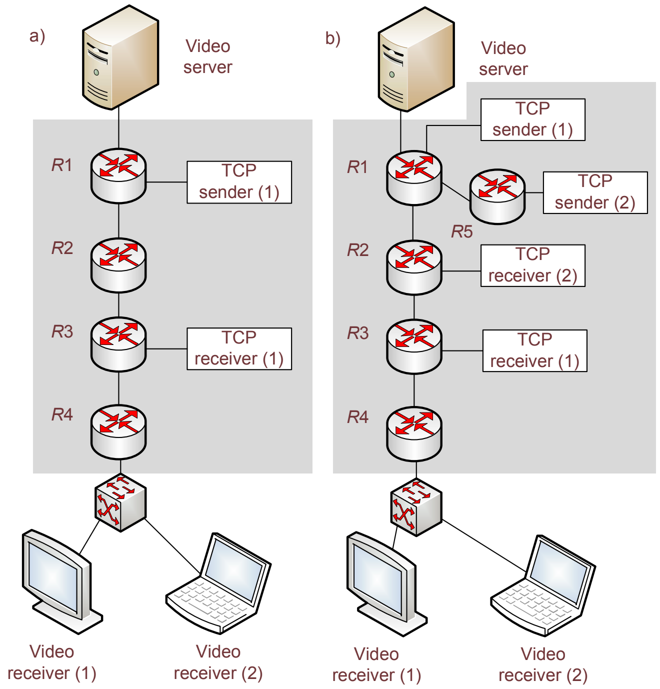

Figure 5. Logical topologies: a) low and medium aggressive TCP, b) high aggressive TCP. Emulated network is marked in gray.

The video transmission was carried out using the RTP/UDP/IP protocol stack, while the TCP/IP protocol stack was used for transmission of elastic traffic. TCP senders were connected to the nearest router (*R*1 or *R*5) via a 65 Mbps link. Throughput of the link between routers *R*1 and *R*5 was set to 200 Mbps.

C. TCP Aggressivity Variants and Simulation Scenarios

Experiments were carried out for three levels of aggressiveness of elastic (based on the TCP protocol) traffic. As a result, we obtained three groups of experiments (Table II): group 1 – lowly-aggressive elastic traffic, group 2 - medium-aggressive elastic traffic, group 3 - highly-aggressive elastic traffic.

TABLE III. RESERVATION SETTINGS.

Scenario	QoS guarantees for HD video transmission	
	QoS guarantees	***Reservation settings***
s1	Best effort	No settings
s2	The classic RSVP protocol (no extensions)	Peak bit rate (well-dimensioned, overestimated reservation)
s3		150% of target bit rate (medium-dimensioned reservation, instantaneously underestimated)
S4	Proposed extensions for the RSVP protocol	Dynamic

For the above variants, 4 basic scenarios were defined. In the scenario s1, HD video transmission was carried out with no QoS guarantees, according to the simplest delivery method - the best effort. In the case of scenarios s2 and s3, quality is assured using the typical RSVP protocol. Scenario s4 uses the proposed, dynamic reservation method. Settings of the scenario parameters are shown in the Table III.

V. RESULTS

Our experiments allow the testing of both proposed RSVP extensions and the classic RSVP with two popular reservation settings (Table III). We compare them with a typical best effort transmission. Results show how both experimental and typical solutions to QoS assurance behave in the presence of HD video traffic which compete for bandwidth with the TCP flow(s) of different aggressiveness.

Figure 6 compares results obtained for HD video competing for bandwidth with lowly aggressive TCP flow. As we can see in the Figure 6a, the basic (best effort) service is not able to properly transfer HD video stream in the real-time. In the case of 5 of 8 video clips, packet error rate (PER) of HD video stream was close to 0.2%. The rest of video clips have obtained PER of about 0.1%. Due to small aggressiveness of the TCP, which is not able to utilize all available bandwidth, link utilization fluctuates from about 80% to about 90%.

Usage of the RSVP protocol improves the QoS of transmitted HD video stream significantly. If the reservation was set to peak bit rate, only ¼ of tested HD video transmissions have PER larger than 0,05%. In many cases packet error rate is equal to zero – i.e. no packets were lost. However, the improvement of the QoS was done at the cost of link utilization (Figure 6b). The link utilization ranges from about 40% to about 60% (typical value - 50%), while best effort transmission gives link utilization of 80%-90%.

After the usage of proposed extensions to the RSVP protocol we observed an increase in link utilization, (70%-90% instead of 40%-60% observed for the RSVP without extensions), which become closer to the one observed for the best effort service (80-90%). Usage of extended RSVP gives perfect QoS (PER equal to zero) in the case of all tested video sequences.

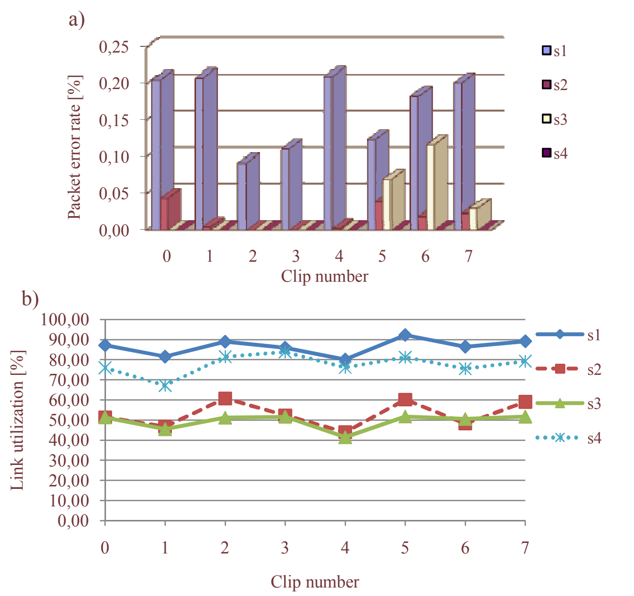

Figure 6. Scenarios of group 1: a) packet error rate of video stream, b) link utilization. Clip numbers: 0 – Aspen, 1 – RedKayak, 2 – WestWindEasy, 3 – RushFieldCuts, 4 – SnowMnt, 5 – SpeedBag, 6 – TouchdownPass, 7 – ControlledBurn.

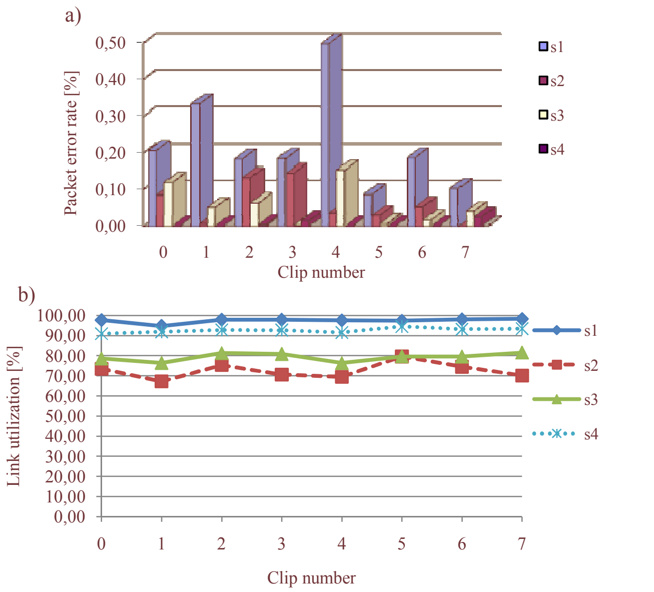

Figure 7. Scenarios of group 2: a) PER of video stream, b) link utilization.

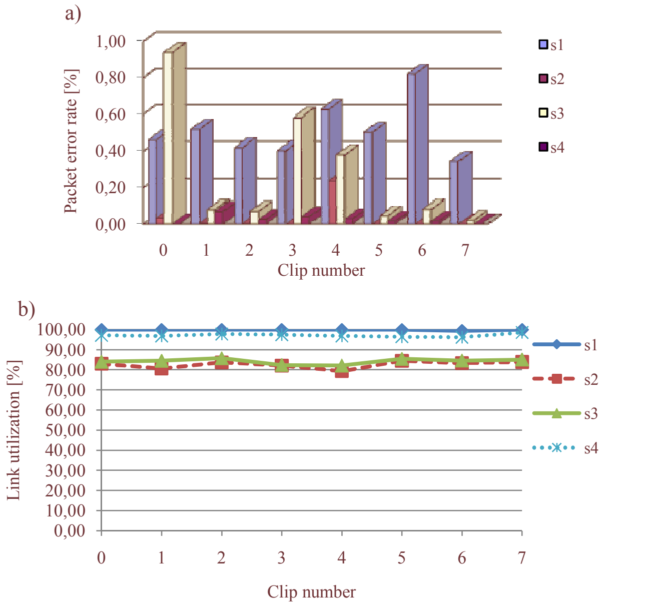

Figure 8. Scenarios of group 3: a) PER of video stream, b) link utilization.

Figure 7 depicts the situation when HD videos share the same link with a medium-aggressive TCP flow. The more aggressive TCP increases link utilization. In the case of the best effort, link utilizations were from 80%-90% to 95-98%. Better link utilization is achieved at the expense of the QoS. Observed packet error rates rises from 0,1%-0,2% to 0,1%-0,5%.

Usage of the RSVP improves QoS, although absolute values of PER are larger than in the case of competition with a lowly-aggressive TCP. PER equal to zero was observed only once. An increase of aggressiveness of the TCP causes an increase of link utilization, from 40%-60% to 70%-80%. Usage of the extended RSVP gives link utilization at the level of 91%-95%. In the case of the extended RSVP we observe the smallest worsening of QoS when compared with the graphs for scenarios of group 1. In 5 of 8 cases PER=0 and in the rest we observe loses of a few packets only. In the worst case (clip 7 – ControlledBurn), 11 of 43750 packets were lost.

Highly-aggressive TCP deepens these tendencies (Figure 8). Best effort service gives link utilization close to 100%, but PER ranges from about 0,3% to about 0,8%. Instantaneous overestimation of reservations caused that link utilization to be at the level of 80%-85% when the RSVP is used. Large PER observed in Figure 8a for scenario s3 are caused by instantaneous underestimation of reservations. Traffic, which can't be sent via the reserved channel, is transmitted using the best effort service, where meets with highly-aggressive TCP. As a result, a small stream of underestimated traffic is sometimes not able to compete with the TCP (e.g. clip 0 in Figure 8a).

The Proposed extension to the RSVP allows the achievement of both high link utilization, close to the one observed for the best effort service, and a very low PER of the video stream. Moreover, results of the Mean Opinion Score (MOS) tests, not included in this paper, show that the proposed solution allows the achievement of a satisfactory quality of experience (QoE)[8].

VI. Conclusions

This paper proposes a new and simple method of dynamic bandwidth reservation, applied as an extension to the RSVP protocol. This method is based on a prior knowledge of the transmitted stream, acquired from the sender buffer of the sender application. The extension uses the additional message which carries data necessary to change the reserved network resources. Experiments, carried out in an emulated multi-node heterogeneous network, show that the proposed extension assures both satisfactory QoS for HD video (comparable with the well-dimensioned classical RSVP reservations), and good link utilization (comparable with the best effort service).

References

[1] F. Boavida, E. Cerqueira, R. Chodorek, M. Grega, M. Leszczuk, Z. Papir, P. Romaniak, and C. Guerrero, "Benchmarking the quality of experience of video streaming and multimedia search services: the content network of excellence," Przeglad Telekomunikacyjny, Wiadomosci Telekomunikacyjne, vol. 81, no. 8-9, 2008..

[2] J. Asghar, F. Le Faucheur, and I. Hood, "Preserving video quality in IPTV networks," Broadcasting, IEEE Transactions on, vol. 55, no. 2, pp. 386–395, June 2009.

[3] T.H. Szymanski and D. Gilbert, "Internet multicasting of IPTV with essentially-zero delay jitter," Broadcasting, IEEE Transactions on, vol. 55, no. 1, pp. 20–30, March 2009.

[4] K. Fall and K. Varadhan, "The ns Manual," January 6, 2009, http://www.isi.edu/nsnam/ns/.

[5] D. Mahrenholz and I. Svilen, "Real-Time Network Emulation with ns-2," Proceedings of The 8-th IEEE International Symposium on Distributed Simulation and Real Time Applications, Budapest Hungary, October 21-23, 2004.

[6] D. Mahrenholz and I. Svilen, "Adjusting the ns-2 Emulation Mode to a Live Network," KiVS'05, Kaiserslautern Germany, 28.02. - 03.03.2005.

[7] VQEG, The Video Quality Experts Group, http://www.vqeg.org/.

[8] R. Chodorek and M. Leszczuk, "QoE validation of a RSVP protocol extension enabling efficient resource reservation for aggregated traffic in heterogeneous IP networks," Second International Workshop on Quality of Multimedia Experience QoMEX 2010, Trondheim, Norway, June 21–23, 2010.