Providing QoS for High Definition Video Transmission Using IP Traffic Flow Description Option

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 Information Technology
Kielce University of Technology
Kielce, Poland
e-mail: a.chodorek@tu.kielce.pl

Abstract — **A real-time multimedia transmission requires the QoS guarantees to preserve the real-time character of the stream. Nowadays, we have several methods for QoS provisioning in IP networks. In this paper we propose a novel method for dynamic network resource allocation using the IP Traffic Flow Description option. Our solution deals with the problem of the trade-off between link utilization and the quality of real-time service. We tested the proposed method using the emulation system, which consists of a streaming server, a network emulator and video receivers. As network emulator, the Berkeley's ns-2 simulator (working in emulation mode) was used. The streaming server and the video receiver were built on the base VLC software. The results of the experiment performed in the test environment show that the proposed solution ensures both satisfactory QoS for HD video and good link utilization.**

Keywords— QoS, network emulation, HD video streaming, heterogeneous IP network, IP traffic flow description option

I. INTRODUCTION

The transmission of high definition video consumes a relatively large amount of network resources and requires a stringent Quality of Service (QoS) [1]. In practice, it cannot be carried out in a heterogeneous network with any sort of QoS assurance [2]. Although in the case of modern local area networks (as for example 802.11) build-in mechanisms are usually good enough to deliver a data stream with the required QoS, in the case of networks larger than LANs specialized architectures are needed. In today's Internet we can use a few well known QoS architectures, such as the Integrated Services (IntServ), the Differentiated Services (DiffServ) or the Flow-Aware Networking (FAN).

Generally, QoS assurance is based on the reservation of network resources. The most important protocol for resource reservation in the Internet is the Resource ReSerVation Protocol (RSVP) [1]. The RSVP is a signaling protocol, used to convey knowledge about QoS-protected transmission (and, as a result, about resources that should be allocated) to intermediate nodes [3]. The protocol was developed as an important part of the IntServ mechanism. Usage of RSVP allows a distribution system to achieve satisfactory QoS [1][3] at the cost of lower link utilization than is offered by the best effort service. Therefore, network resources typically aren't optimally utilized 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 simply define. This problem is especially noticeable in the case of streaming HD video because of its large bit rates compared to the typical bit rates of aggregated traffic.

In this paper we propose a novel method of dynamic resource allocation which deals with the problem of a trade-off between link utilization and quality of real-time service. The proposed method dynamically changes resource reservations on the basis of knowledge about transmitted traffic. Because the RSVP signaling is static and cannot transmit dynamic information about forthcoming traffic, in our solution the IP Traffic Flow Description option [4] was used for QoS signaling.

The IP Traffic Flow Description option is set in the sender according to heuristic analysis of the sending buffer of the video streaming application. Because such a buffer is a typical element of streaming applications, the proposed solution doesn't interfere with the standard streaming mechanism.

The paper is organized as follows. Section II presents the new IP option (the IP Traffic Flow Description option) and its applicability to the description of video streams. Section III is devoted to the proposed resource allocation using the IP traffic flow option. Section IV presents the emulation environment used for the analysis. Section V describes experiments and shows obtained results. Section VI concludes the paper.

II. THE IP TRAFFIC FLOW DESCRIPTION OPTION AND ITS APPLICABILITY TO DESCRIBE THE VIDEO STREAM

Many real-time applications have or may have at their disposal information about the volume of traffic that will be sent in the near future. This data may be derived, for example,

The work presented in this paper was supported by the Statutory Activities of KT AGH (Contract No. 11.11.230.018).

from the video compression process or transmission buffer. They also can be obtained from the predictor of the video traffic built into the application [6][12].

a)

0 7	8 15	16 31
100xxxxx	Len	Flags
Next Data		
Next Time		

b)

0 7	8 15	16 31
Next Hdr	Len	Flags
Next Data		
Next Time		

c)

16 24	25	26	27	28	29	30	31
Res	D	M	B	F	L	S	E

Fig. 1. IP Traffic Flow Description option [4]: a) IPv4, b) IPv6, c) Flags field

The IP Traffic Flow Description option has the format shown in Fig. 1 [4]. The most important fields describing traffic are Next Data and Next Time. The field Next Data (32 bit) conveys the size (in bytes) of data sent in the near future. The field Next Time (32 bit) conveys time (in milliseconds) it will take to send the data that was included in the field Next Data. The field flags determines the format of Next Data and the properties of the transmitted data (Fig. 1c). If the transmitted data is non elastic video traffic the flag S must be set to 1.

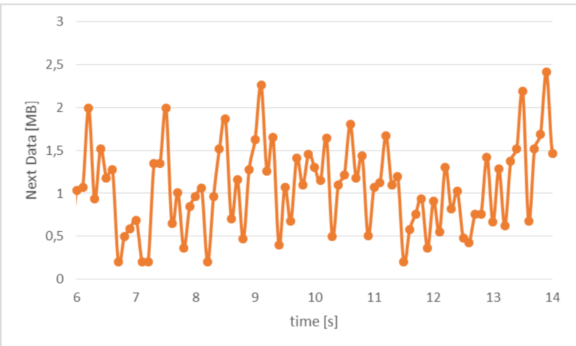

Fig. 2. Values of the Next Data field for clip Aspen

Below is an example of the use of the IP Traffic Flow Description option during HD video streaming. Video streaming applications analyze the sending buffer. Therefore flag B must be set to 1 (Next Data field is set on the basis of buffer analysis) [4]. The sending buffer is set to 300 ms. Therefore the Next Time is set to 300 ms. During transmission the sum of all packets stored in the buffer is continuously calculated. The sum is stored in the Next Data field. Next Data field values for the IP Traffic Flow Description option for the Aspen clip are shown in Fig. 2.

The data send by the application is identified as a unique flow of packets defined by unique packet parameters. In the IP packets, certain IP header fields are used to define a flow, including IP addresses and some other fields, e.g. IPv6 flow label. In IPv4 a flow may be defined by selected IPv4 header fields and transport protocol fields. Typically, it could be a standard 5 tuple (source IP address, source port, destination IP address, destination Port and transport protocol ID). In IPv6 it is enough to define flow by source IP address, destination IP address and IPv6 flow label.

The new IP option presented in [4] allows the distribution of information about future values of traffic being sent within a given flow.

III. PROVIDING QOS USING THE IP TRAFFIC FLOW DESCRIPTION OPTION FOR VIDEO STREAMS

Providing QoS and at the same time providing a good utilization of network resources requires detailed information about the demand for network resources for the QoS services. The video services may have information about how much traffic will be sent in the near future. The information about how much traffic will be generated by the video service can be sent to the network nodes and used for dynamic allocation of resources in the network. The allocation may be carried out for example by extensions to RSVP[1] protocol [5]. Another solution is to use the IP Traffic Flow Description option [4].

The IP packets are sent to the destination(s) by typical QoS capable routers. The QoS routers often implement the Weighted Fair Queuing (WFQ) algorithm. The router has the M queues, the last queue of the index M is for best-effort traffic. The number of queues depends on the network QoS strategy defined by the router configuration. In this paper we analyzed the simplest QoS strategy which is for $M = 2$, where in the router there are only two queues. The first queue serves to stream traffic that has a defined QoS. The second queue serves other traffic. The most advanced strategy creates a separate queue for each stream. In the proposed solution every stream or group of streams is assigned to a queue at index j with weight w_j. This weight is set according to the required QoS parameters.

In the proposed solution routers collect information about each data stream from the IP Traffic Flow Description option. This data is stored in the Traffic Flow Block (TFB) data structure. The TFB structures for each stream are defined in the output interface. Access to the TFB structure is realized through the stream index i. For each stream in the TFB structure the time Ts_i of the first appearance of the stream is saved. Each entry in the structure of the TFB has its validity defined based on the time (field Next Time) specified in the IP Traffic Flow Description option and entry time it went live defined in the configuration of the router. If, before the expiry of the validity time of the TFB entry the router does not receive the next IP packet of the stream with the IP Traffic Flow Description option, the entry is removed. Each stream in the TFB structure has the required bandwidth R_i stored for the stream based on the Next Date and Next Time fields.

[1] An early concept of extensions to the RSVP protocol and the QoS analysis of that concept was presented by authors in [11]. The QoE evaluation of that concept was presented in [13].

The bit `s_be` in the TFB structure denotes the stream for which QoS guarantees are excluded. This bit is set for these streams which declare a desired bandwidth exceeding bandwidth of the output interface where $R_i > R$. It is also used, as described in the next section, in the algorithm which determines weights for the WFQ queue. Streams excluded from the QoS guarantees are classified to the last queue (index *M*) destined for best-effort traffic.

In the proposed solution, the router calculates the total bandwidth requirements for traffic when the router must guarantee an adequate QoS level. For each queue *j* the total bandwidth is determined by the streams assigned to it. The router for calculations include only those streams that have `s_be` bit cleared.

If we uses only two queues the total value of estimated bandwidth *R'* of the QoS queue is calculated from the relationship:

$$R' = \sum_{i=1}^{N_j} R_i \tag{1}$$

If the estimated bandwidth *R'* is larger than or equal to the output interface throughput *R* the router must exclude one of the streams from the QoS queue. In the proposed solution the youngest stream is rejected, i.e. the stream for which the time *Tsd* satisfies the relationship:

$$Tsd = \min_{0<i\leq N} Ts_i \tag{2}$$

The rejected stream is marked by setting the `s_be` bit in the structure of TFB.

After eliminating one of the streams, the router again estimates the required bandwidth. If the estimated bandwidth *R'* is larger or equal than the output interface throughput *R* the procedure which exclude the youngest stream is repeated. The procedure is repeated until the estimated total bandwidth *R'* is smaller than the throughput of the output interface *R*. After the process of estimation and optimization of the bandwidth *R'* ends the weight will be calculated according to formula:

$$w = \frac{R'}{R} \tag{3}$$

The queues will be correctly weighted and the list of streams for which the router is able to provide the QoS parameters (correct number of streams marked by bit `s_be`) will be defined in the router periodically in accordance with the time configured in the router. This correction may also be taken when the traffic parameters of the stream exceed the level defined in the configuration.

IV. TESTING ENVIRONMENT

The proposed solution was tested in a heterogeneous, multiservice environment, where real-time video traffic competes for bandwidth with bulk data flows. The test environment consists of video server (*SM*), TCP server (*STCP*), network emulator, a set of video receiver(s) (*RMi*, $i = 0,\ldots, N$) and a set of TCP clients (*RTCPi*, $i = 0,\ldots, K$).

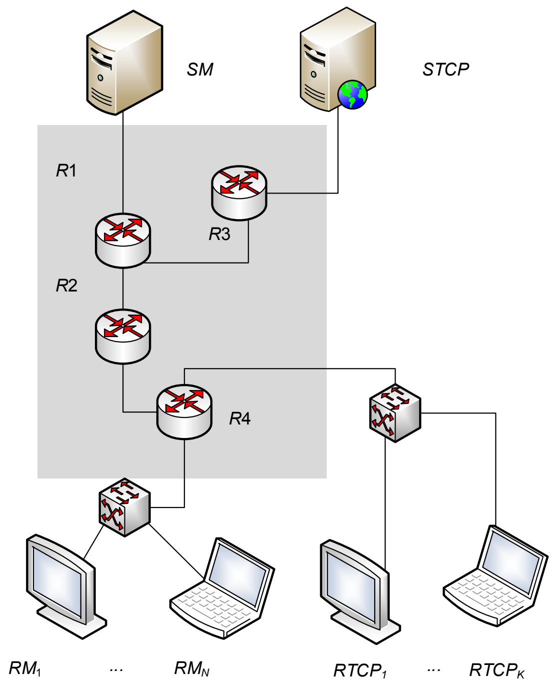

Fig. 3. Test environment

The test emulated network (market in gray on Figure 3) consists of four routers (from *R*1 to *R*4). In all experiments the throughput of links inside the emulated network was large enough to ensure live video transmission – 65 Mbps between each routers. External connections (between emulated network and real word) are build using Gigabit Ethernet technology (network card, switch).

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 (bulk data flows).

The video streaming server *SM* is a computer with the Linux operating system. The Linux kernel included our implementation of IP Traffic Flow option. The VLC media player[2] with our extension to support IP Traffic Flow Description option is a video streaming application running on the server *SM*. Video receiver(s) are computers with Linux operating systems and VLC media players as applications receiving the video stream.

The Berkeley's ns-2 simulator working in an emulation mode [10] was used as the network emulator. The emulation mode enables 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

[2] http://www.videolan.org/vlc/

simulated network environment, to inject them into the real network.

The emulator, used in our research, was supplemented by extensions developed at the University of Magdeburg (Germany) [7][8]. Supplementary software included extensions for the real-time scheduler module and the modification of modules enabling co-operation with a real network. To enable emulation of real-time HDTV video streams, we supplemented the emulator with our original extensions for a real-time scheduler module and interface module.

The video server (SM) was built on a high performance PC equipped with an Intel® multicore processor and a Gigabit Ethernet card. The network emulator (*NetSrv*) was built on the basis of the Intel® Server Board platform with a dual Gigabit Ethernet. The *NetSrv* was equipped with two Intel® Xeon® processors, 16 GB of RAM memory. As end systems highly efficient PC computers were used. The clients (both video and TCP) were equipped with Intel® multicore processors and a Gigabit Ethernet interface.

In this paper for multimedia content High Definition Television (HDTV) video sequences were used for the experiments. The video content is publicly available at The Video Quality Experts Group (VQEG) site [9], at URL ftp://vqeg.its.bldrdoc.gov/HDTV/NTIA_source/. 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.

The HDTV collection consists of 8 clips (*Aspen, RedKayak, WestWindEasy, RushFieldCuts, SnowMnt, SpeedBag, TouchdownPass* and *ControlledBurn*), each lasting 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 the WestWindEasy clip) sudden scene changes.

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.

Four (s1 to s4) basic scenarios were defined. In the s1 scenario, 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. The scenario s2 appled peak bit rate (well-dimensioned, overestimated) reservations. The scenario s3 applied a 150% of target bit rate (medium-dimensioned) reservation, instantaneously underestimated. Scenario s4 used the proposed QoS assurance for High Definition video stream transmission using the IP Traffic Flow Description option.

Experiments were carried out at many levels of network load. In the experiments we changed the numbers of the concurrent TCP connections. As a result, our network was loaded at a low, medium and high level. As a reference, we tested an unloaded network, where only video transmission took place.

V. RESULTS

Proposed dynamic resource allocation using IP Traffic Flow Description option was tested for HD video, which required relatively large network resources. We tested QoS assurance for HD video transmission with simultaneously transmitted elastic traffic in the same link. We also analyzed link utilization when different scenarios were used.

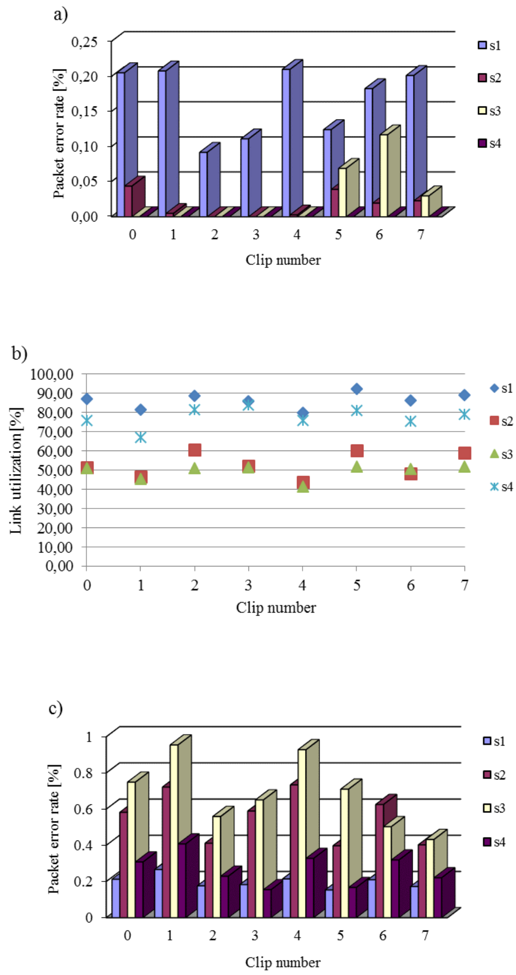

Fig. 4. Scenarios for HD video competing for bandwidth with one TCP connection: a) packet error rate of video stream, b) link utilization, c) packet error rate of TCP connection. Clip numbers: 0 – Aspen, 1 – RedKayak, 2 – WestWindEasy, 3 – RushFieldCuts, 4 – SnowMnt, 5 – SpeedBag, 6 – TouchdownPass, 7 – ControlledBurn.

Figure 4 compares results obtained for HD video competing for bandwidth with one TCP connection. As we can see in the Figure 4a, the best effort service is not able to properly transfer HD video stream in the real-time. The quality of HD video is not acceptable. For the 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%. One TCP connection is not able to utilize all available bandwidth. Link utilization fluctuates from 80% to 90%.

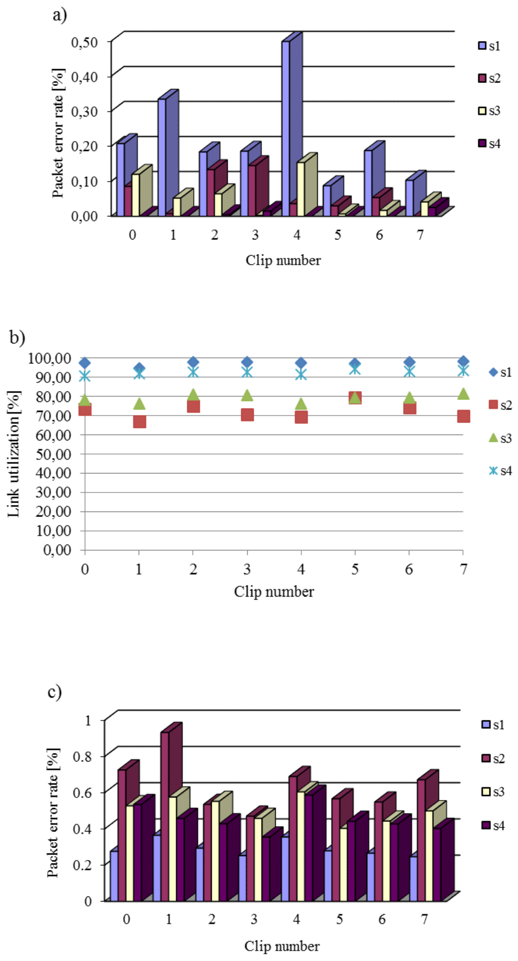

Fig. 5. Scenarios for HD video competing for bandwidth with two TCP connections: a) packet error rate of video stream, b) link utilization, c) packet error rate of TCP connections.

The QoS of transmitted HD video stream significantly improves when the network uses the RSVP protocol (scenario s2 and s3 in Figure 4a). In scenario s2, only ¼ of tested HD video transmissions have PER larger than 0.05%. However, the improvement of the QoS was done at the cost of link utilization (Figure 4b) which not exceeds 60%.

The proposed QoS assurance for HD video stream transmission using IP Traffic Flow Description option was tested at scenario s4. We observed an increase in link utilization (70%-90% instead of 40%-60% observed for the RSVP – Fig. 4b) and perfect QoS (PER equal to zero) in the case of all tested video sequences (Fig.4a). For scenario s4 the PER for TCP connection (Fig. 4c) is also smaller than observed for scenarios s2 and s3 and is comparable with results from scenario s1 (best effort).

Bulk data transfer using two independent TCP connections, which took place in the background of the HD video transmission, is shown in Fig. 5. Although multiplication of TCP flows increases the link utilization, we observe a significant deterioration of the quality the transmission for the scenario s1 (best effort). Application of static reservations based on the RSVP protocol (s2 and s3 scenarios) significantly improves QoS.

The growth of the TCP traffic increases the link utilization from 40% - 60% to 70% -80%. Application of the proposed solution (scenario s4) improves that result and gives link utilization at the level of 91% - 95%. In the case of scenario s4, in 5 of the 8 cases PER is equal to 0, and for the rest we lose only a few packets (for the worst case - clip 7 - 11 packages of more than 43 thousands). The PER for TCP connections (Fig. 5c) is comparable to results for the one TCP connection (Fig. 4c).

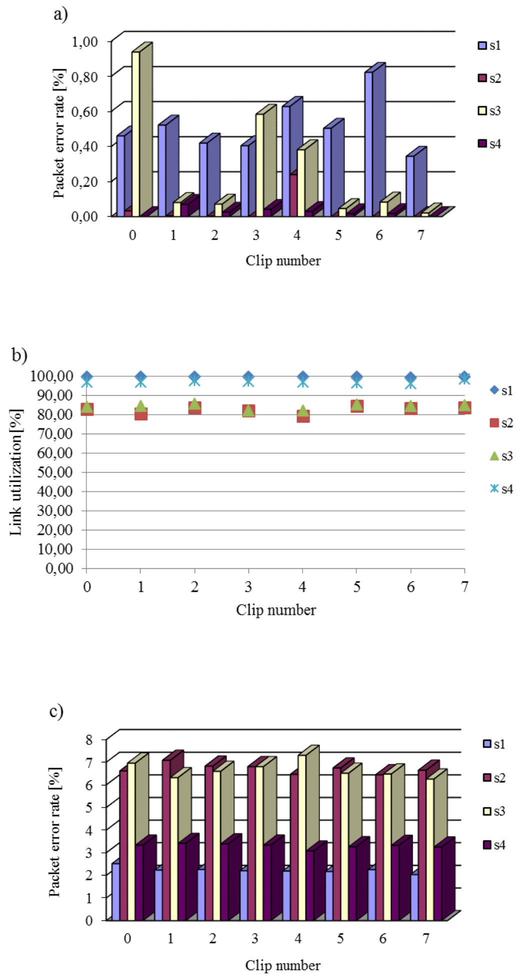

Fig. 6. Scenarios for HD video competing for bandwidth with three TCP connections: a) packet error rate of video stream, b) link utilization, c) packet error rate of TCP connections.

Figure 6 depicts the case of bulk data transmission using three TCP connections, where one of the TCP flows has a large transmission window (this TCP connection, practically, was not flow controlled). As we can see in the figure, in the presence of a large transmission window the tendency observed in Fig. 5 is deepening. The strategy s1 (best effort) provides a link utilization almost 100%, but the observed PER is in the range of from about 0.3% to about 0.8% (Fig. 6a). The usage of the RSVP protocol (s2 and s3 strategies) causes the losses to decrease for the transmission of the HD video (with the exception of the clip 0 and 3 for the strategy s3), but the link utilization is at 80% - 85% (Fig. 6b). The proposed solution (strategy s4) gives for the transmitted HD video very small losses (Fig. 6a), with good utilization of the links at more than 95% (Fig. 6b). Losses observed for the TCP connections (Fig. 6c) are very large and for the reservations through the RSVP protocol (s2 and s3 strategies) reach 7%.

VI. CONCLUSIONS

This paper proposes a new, simple way to ensure QoS for streaming HD video using the IP Traffic Flow Description option. The proposed solution uses information about the future value of the transmitted stream extracted from the sender buffer. This information is transmitted from the transmitter to the intermediate node using the IP Traffic Flow Description option.

Experiments carried out in an emulation environment, show that the proposed solution provides satisfactory QoS for HD video (comparable with typical reservations that use the RSVP signaling) and good link utilization (comparable to the best effort).

References

[1] J. Asghar, F. Le Faucheur, I. Hood, “Preserving video quality in IPTV networks,” IEEE Transactions on Broadcasting, 2009, 55, (2), pp. 386–395, doi: 10.1109/TBC.2009.2019419

[2] A. Chodorek, R.R. Chodorek, A. Krempa, “An analysis of elastic and inelastic traffic in shared link,” Conference on Human System Interactions, May 2008, pp.873-878, doi: 10.1109/HSI.2008.4581558

[3] F. Pana, F. Put, “A Survey on the Evolution of RSVP,” IEEE Communications Surveys & Tutorials, 2013, 15, (4), pp. 1859-1887, doi: 10.1109/SURV.2013.021313.00107

[4] R.R. Chodorek, “An IP option for describing the traffic flow,” IETF Internet Draft draft-chodorek-traffic-flow-option-03, 2015

[5] R.R. Chodorek, A. Chodorek, “RSVP Extensions for Dynamic Reservation,”, IETF Internet Draft draft-chodorek-tsvwg-rsvp-dynamic-resv-00, 2015

[6] A. Chodorek, “Prediction-based dynamic QoS assurance for multicast multimedia delivery,” High-Speed Networks and Multimedia Communications: 6th IEEE International Conference HSNMC 2003, Estoril, Portugal, July 23-25, 2003, Proceedings. Vol. 6. Springer, 2003, pp. 128-135, doi: 10.1007/978-3-540-45076-4_13

[7] D. Mahrenholz, 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

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

[9] VQEG, The Video Quality Experts Group, http://www.vqeg.org/, accessed March 2015

[10] K. Fall, K. Varadhan, “The ns Manual,” November 4, 2011, http://www.isi.edu/nsnam/ns/, accessed March 2015

[11] Robert Chodorek, Agnieszka Chodorek, An Analysis of QoS Provisioning for High Definition Video Distribution in Heterogeneous Network, 14th International Symposium on Consumer Electronics (ISCE 2010), 7-10 June 2010, Braunschweig, Germany

[12] A. Chodorek, “A fast and efficient model of an MPEG-4 video traffic based on phase space linearised decomposition,” Proc. of 14th European Simulation Symposium ESS'2002, 2002

[13] Robert Chodorek, Mikolaj 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), 21-23 June 2010, Trondheim, Norway