An Analysis of Elastic and Inelastic Traffic in Shared Link

Agnieszka Chodorek†, Robert R. Chodorek‡, and Agata Krempa‡
†Kielce University of Technology, Department of Telecommunications and Photonics, Kielce, Poland
‡AGH University of Science and Technology, Department of Telecommunications, Kraków, Poland

Abstract — **Nowadays, we observe rapid growth of multimedia transmissions in the Internet. Multimedia are typically carried out using the RTP protocol, while the rest of Internet traffic is transmitted via the TCP protocol.**

In the paper the problem of coexistence of elastic and inelastic traffic is presented. In contrast to other papers, which analyze this issue as the problem of TCP-unfriendliness of the RTP protocol, we focus on this coexistence from multimedia point of view.

Keywords — **multimedia transmission, RTP, TCP.**

I. Introduction

INTERNET traffic consists of data sent by different kinds of applications such as web traffic, FTP traffic and real-time multimedia traffic (voice and video). Different applications use different transmission protocols to send their data. FTP and web traffic are sent using TCP protocol. Voice and video are sent using RTP/UDP protocol suite. Sending different kinds of data simultaneously can result in one transmission having negative effect on the other one. It is highly accurate in case of mixing responsive flows such as TCP and nonresponsive flows such as RTP or UDP. In that scenario RTP/UDP flows tend to suppress TCP flows resulting in disproportional use of bandwidth. TCP senders can be even throttled down by RTP/UDP traffic as RTP/UDP senders do not reduce their transmission speed in time of congestion. This problem is especially relevant as the percentage of the RTP/UDP traffic increases. This problem was named TCP-unfriendliness [1] and TCP-Friendly Rate Control [2] was proposed as a solution. Considering TCP flows, it’s reasonable to ask all flows (TCP and RTP/UDP) to fairly share available bandwidth. But, on the other hand, reduction of RTP/UDP transmission rate can cause real-time application to be no longer real time. From the services point of view, forcing RTP/UDP flows to adapt to rate of TCP flows is unreasonable. Some authors suggest that balance between elastic and inelastic traffic should be achieved not on a per-flow basis but on the aggregate level [3], [4]. Also some work were presented showing that not all of real-time traffic is a threat to TCP traffic [5].

In the paper we present the issue of elastic and inelastic traffic in shared link. Section 2 of this paper presents survey of elastic and inelastic traffic. Section 3 describes experiments that were made. In Section 4 simulation results are presented. Section 5 concludes this paper.

II. Elastic and inelastic traffic

Elastic traffic is not sensitive to delay. Figuratively speaking, it can spread in time. This kind of traffic is associated with applications that send their data using TCP protocol. They are application such as FTP, WWW and e-mail. They direct to transport protocol a continuous set of data (file, message, e-mail or web page) and the transmission rate of this data depends on transport protocol mechanisms and network conditions. Because this transmission does not have time borders (e.g. file transfer can last one minute as well as 10 seconds) it doesn’t have to meet real time conditions.

Because real time conditions don’t have to be met, elastic traffic is invulnerable to delay and jitter. It also doesn’t have minimum bandwidth requirements (but high throughput is desirable). But it requires correct data transmission, which is achieved by reliable transport protocol (as TCP). Because of that there is no need to employ quality of service techniques. Default best-effort service is enough.

Inelastic traffic is vulnerable to delay – it cannot be spread in time. This kind of traffic is associated with applications, which use RTP protocol. The main services that generate inelastic traffic are VoIP, VoD, IPTV, audio- and videoconference. They direct to transport protocol discrete stream of data (e.g. image from camera in form of video frames sent every 40 ms) and the rate of transmission is determined for encoding and compression method of multimedia data. In case of streaming applications, data should be received in the same rate they are generated, so real time conditions must be met.

In case when elastic and inelastic traffic is sent in shared link, two different kinds of control can result in adverse effects for both kinds of traffic. On the one hand, as is commonly known, inelastic traffic is not TCP-friendly and, in some situations, we can observe collapse of elastic traffic. On the other hand evolution of TCP’s congestion window, in some conditions, can lead to unacceptable high packet loss of RTP flows that compete with TCP flows about bandwidth.

This work is supported by Polish Government under Grant No. N517 012 32/2108 (years 2007-2009).

III. Simulation experiments

Simulation experiments were run in ns-2 discrete event-driven simulator [6] for typical single-bottleneck topology (Fig. 1.).

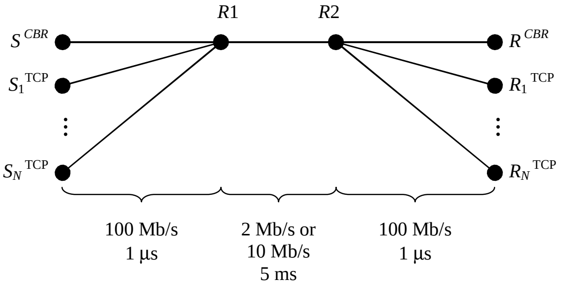


Fig. 1. Network Topology.

Senders, in Fig. 1 represented by symbol S, were connected to router R1. Capacity of the link was set to 100 Mb/s and propagation delay was set to 1 µs. Receivers, in Fig 1 represented by symbol R were connected to router R2 with link with the same parameters. Routers R1 and R2 were connected by link with capacity set to 2 Mb/s or 10 Mb/s and propagation delay set to 5 ms. This link forms a bottleneck of the system.

We were changing queue management method in ingress interface of router R1 during experiments (according to Tab. 1). Management method of remaining queues was set to DropTail with buffer size set to 50 packets.

Table 1: Queue parameters of ingress interface of R1 router.

Parameter	***Experiment symbol***		
	DT	***RED***	***RED&ECN***
Queue management	DropTail	RED	RED
Queue size [packets]	50	50	50
linterm_	-	10	10
thresh_	-	5	5
maxthresh_	-	15	15
q_weight_	-	0.002	0.002
gentle_	-	false	false
setbit_	-	false	true
Congestion notification type	implicit		explicit
Congestion notification method	Gaps in sequence space		ECN bits

Between nodes S^{CBR} and R^{CBR} inelastic traffic was sent (video stream) with constant bit rate (CBR) B (Mbps $\leq B \leq$ 2 Mbps). Video frames were generated every 40 ms. Traffic was sent in two scenarios:

- Bursty traffic (BT),
- Burstless traffic (BLT).

In the case of bursty traffic, all of the packets that are forming a frame, are sent around the same time, forming a burst of packets. This is a typical method of sending multimedia traffic. In case of burstless traffic, burst of packets is spread in time. Because the typical model of CBR traffic (implemented in ns-2 simulator), don't send bursts of packets, it was replaced with our own model of CBR traffic. Number of packets in one burst in relation to bit rate B is shown in Tab 2.

Table 2: Number of RTP packets in one burst.

Bit rate B [Mb/s]	***Size of burst [packets]***	***Bit rate B [Mb/s]***	***Size of burst [packets]***
B = 1 Mb/s	5	B = 6 Mb/s	30
B = 2 Mb/s	10	B = 7 Mb/s	35
B = 3 Mb/s	15	B = 8 Mb/s	40
B = 4 Mb/s	20	B = 9 Mb/s	45
B = 5 Mb/s	25	B = 10 Mb/s	50

Table 3: Parameters of TCP and UDP simulation models.

Parameter	***RTP***	***TCP***
Packet size [B]	1000	1000
Payload size/MSS [B]	960	960
rwnd size [packets]	-	20
Frequency of generation of video frames [Hz]	25	-

Video transmission took place with the use of RTP transport protocol. Packet size of RTP packet was set to 1000 B (Tab. 3).

Between end nodes S_i^{TCP} and R_i^{TCP}, i = 0, 1, ..., N, $0 \leq N \leq 10$, elastic traffic was sent that was generated by FTP application. In experiments we used a built-in FTP traffic generator. As a transport protocol TCP in SACK version was used. TCP protocols parameters were presented in Tab. 3.

As a part of the work, a number of simulation experiments were run, during which we made an analysis of QoS parameters of elastic and inelastic traffic. In case of inelastic traffic following variables were investigated:

- Throughput of RTP flow,
- Packet error rate.

In case of elastic traffic we took into consideration:

- Throughput of TCP flow,
- Packet error rate.

Note that, all of the transmission errors were corrected by error correction mechanism.

IV. Results

In the first experiment we were changing the number of competing TCP flows from to N = 0 to N = 10. Target bit rate of CBR source was set to B = 1 Mb/s, and bandwidth of the bottleneck link was set to 2 Mb/s. Experiment results are presented in Fig. 2 and Fig. 3.

When only RTP/UDP traffic source and one TCP traffic source is present in the network, bandwidth is equally shared between those two flows. Increasing the number of TCP flows, creates a state of heavy load in the network. As congestion occurs and RTP packets are dropped, it allows for TCP flows to occupy more bandwidth (but the mean amount of bandwidth occupied by single TCP flow decrease with increasing number of TCP flows). When tail drop queue management was used and source was sending bursty traffic, more TCP senders caused that more RTP

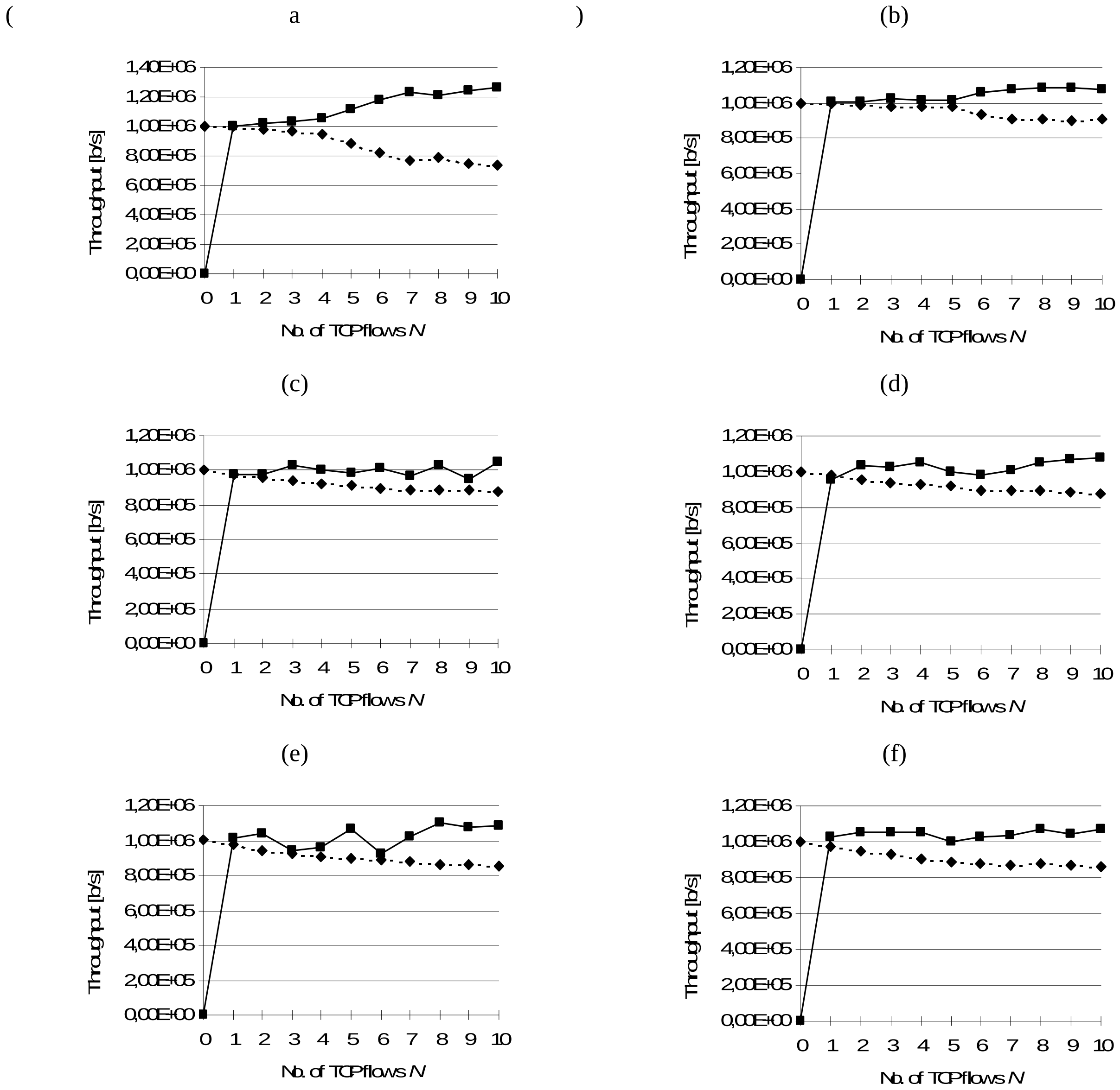

(a) (b) (c) (d) (e) (f)

Fig. 2. Throughput of RTP flow (dotted line) and total throughput of TCP flows (solid line) as a function of number of TCP flows *N*: a,b) DT experiment, c,d) RED experiment, e,f) RED&ECN experiment, a,c,e) bursty traffic, b,d,f) burstless traffic.

packets were lost as whole bursts were dropped in time of congestion. This resulted in lower RTP throughput and allowed TCP flows to win over more of the bandwidth. In case of burstless traffic source the disproportion between throughput of RTP flow and aggregated TCP flows is smaller as less number of RTP packets is dropped. When queue management method was changed to RED less number of RTP packets were dropped so disproportion in throughput of RTP flow and TCP flows is smaller than in case of tail drop queue management and bursty traffic. For RED queue management the results were similar for both bursty and burstless traffic. Results in experiments with RED with ECN signaling mechanism were similar to those with RED queue management with implicit congestion notification, but more RTP packets were dropped when ECN was used.

Taking into consideration that multimedia traffic can tolerate only certain level of errors, tail drop queue management appears to be better for multimedia traffic. Even though RTP packet error rate increase rapidly for bigger number of TCP flows, for four TCP flows errors are yet on the bounds of acceptance in case of bursty traffic. For burstless traffic there can be even five TCP flows and packet error rate will be still on the level of acceptance. In case of RED queue management two TCP traffic source are the limit if we don’t want to exceed the limit of accepted level of errors for multimedia traffic.

In the second experiment we were changing bit rate of CBR source from B = 1 Mb/s to *B* = 10 Mb/s. Number of competing TCP flows were set to *N* = 1, and bandwidth of the bottleneck link was set to 10 Mb/s. Experiment results are presented in Fig. 4 and Fig. 5.

Increasing the bit rate of CBR flow cause the throughput of CBR flow to increase whereas throughput of TCP flows constantly decreases. For burstless traffic, regardless of queue management method, setting bit rate of CBR flow close to link capacity resulted in TCP starvation. Similar effect can be seen for bursty traffic when RED queue

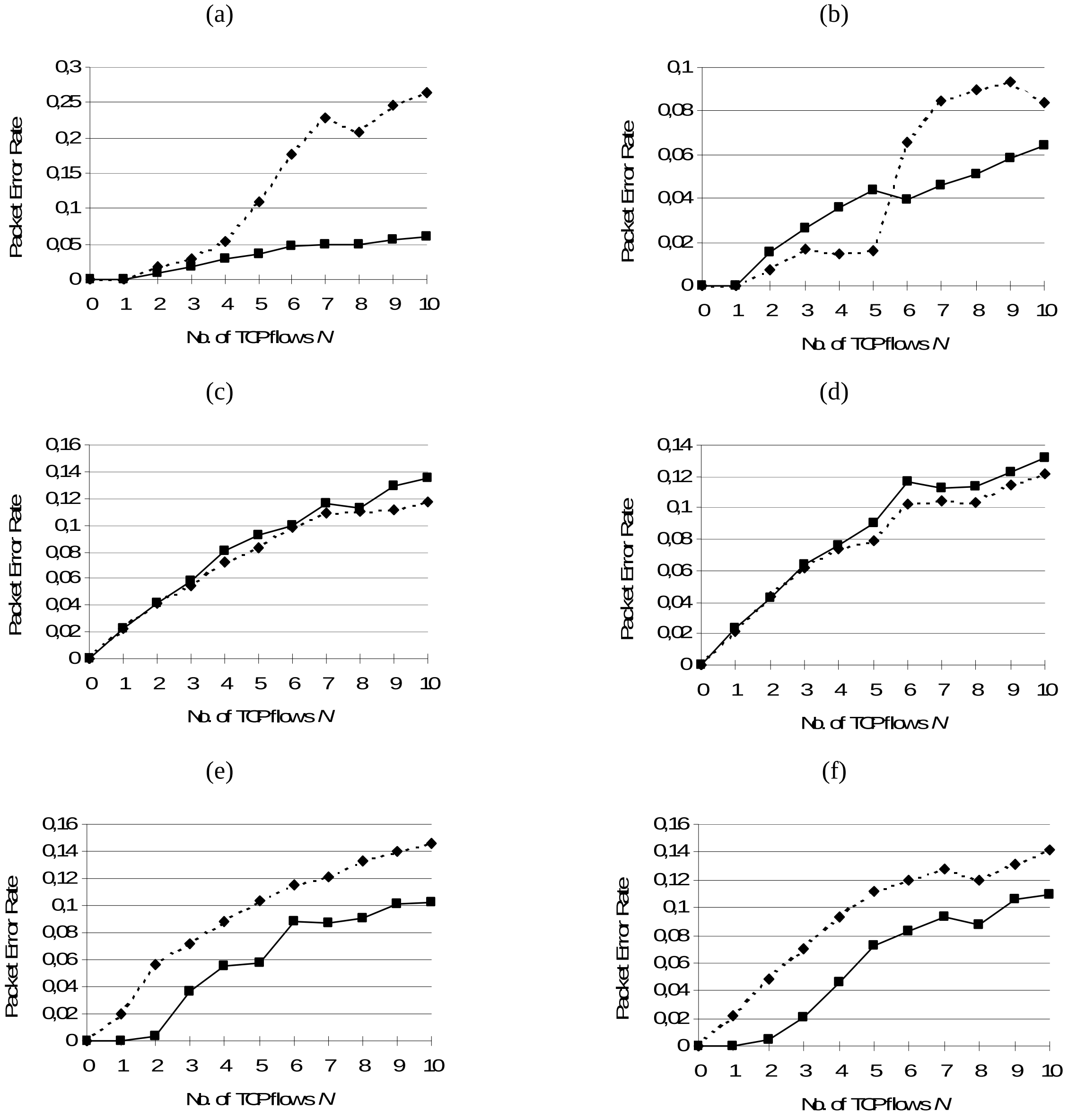


Fig. 3. Packet error rate of RTP flow (dotted line) and summary packet error rate of TCP flows (solid line) as a function of number of TCP flows *N*: a,b) DT experiment, c,d) RED experiment, e,f) RED&ECN experiment, a,c,e) bursty traffic, b,d,f) burstless traffic.

management algorithm or ECN mechanism is used. In all of those cases packet error rate for RTP flow is on acceptable level for CBR traffic source sending data with bit rate up to 9 Mb/s. The only experiment that didn't result in TCP starvation was scenario with tail drop queue management algorithm and CBR source sending bursty traffic. From CBR flow bit rate set to 6 Mb/s, RTP and TCP flows reach a steady state where their throughput is maintained on constant level. Also from this point RTP packet error rate start to increase and instantly reach not accepted level. TCP flows escape starvation because when RTP packet burst is dropped by tail drop algorithm, it opens an opportunity for TCP flows to win over some bandwidth.

Summarizing our experiences we can notice, that the problem of collapse of elastic traffic (which competes for bandwidth with inelastic traffic) can be solved in tree ways. The first one, and the most typical, is to use one of TCP-friendly transport protocols instead of the RTP protocol. However, as we can see in Fig. 2-5, if a network, is well dimensioned for multimedia, RTP will no be a killing protocol for TCP. Moreover, the usage of any TCP-friendly protocol can be dangerous from inelastic traffic point of view, because it can damage real-time character of the RTP flow. Thus, in some situations, alternative solutions are needed. An analysis of such solutions was carried out in he first and the second simulation experiment.

The second way to improve performance of competing elastic and inelastic traffic is to use a queuing management technique, which is more advanced than the tail drop. Typically, the RED queue is used. However, simulations show, that the usage of RED queue will improve coexistence of different kinds of traffic only if one elastic stream competes for bandwidth with one inelastic stream (the second experiment). If several elastic streams compete for bandwidth with one inelastic stream (the first experiment), usage of the RED queue will not improve but

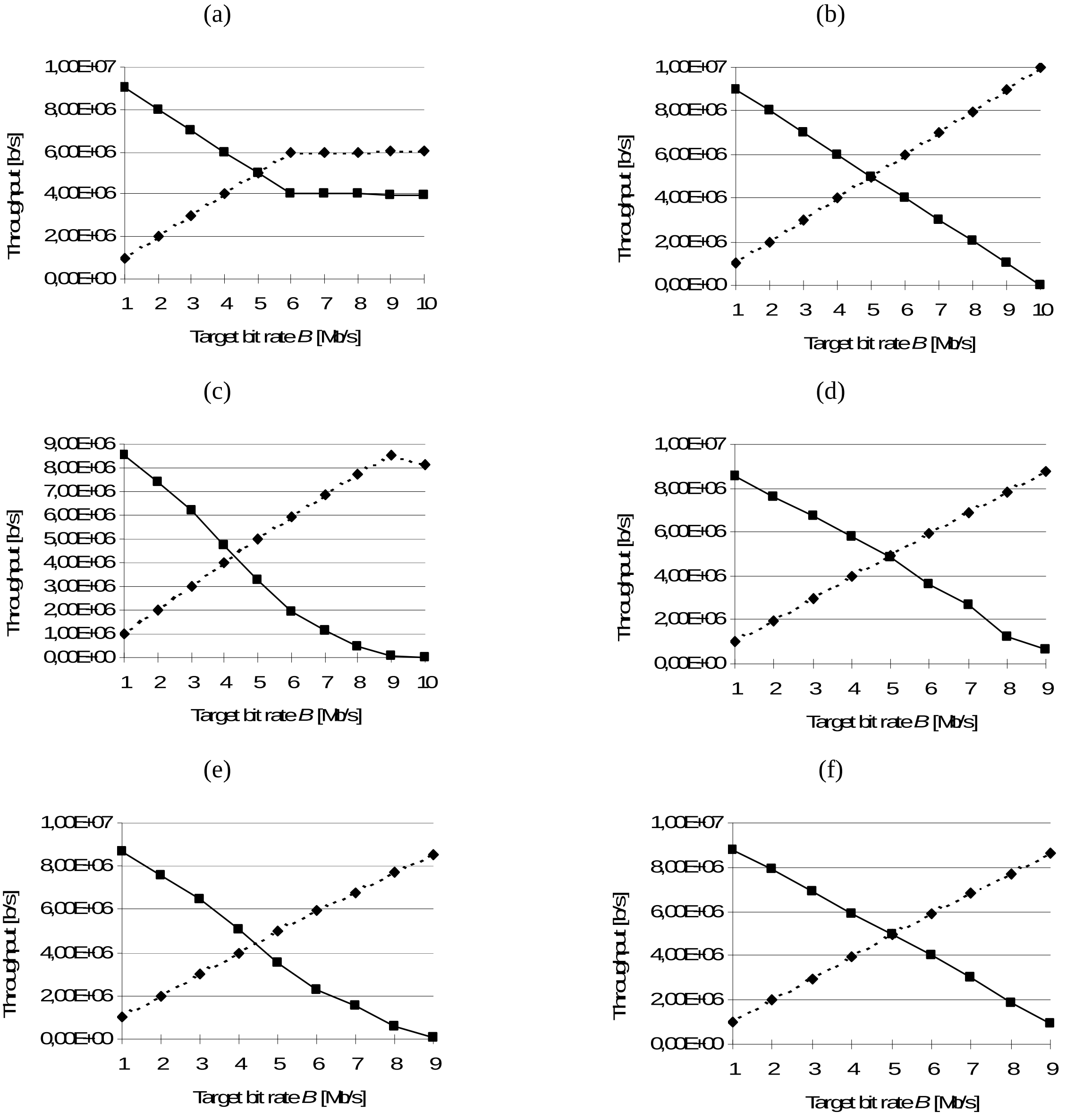


Fig. 4. Throughput of RTP flow (dotted line) and throughput of TCP flow (solid) as a function of target bit rate *B* of CBR stream: a,b) DT experiment, c,d) RED experiment, e,f) RED&ECN experiment, a,c,e) bursty traffic, b,d,f) burstless traffic

worsen coexistence of two kinds of traffic. In he case of the first experiment, if the tail drop queue is used, inelastic stream will achieve limit of acceptance (PER = 5%) when shares link with 4-5 elastic flows. If we use RED queue instead of tail drop, 1-2 elastic flows will be quite enough. Results of the first experiment are especially important, because RED queuing technology is widely used to mitigate the problem of traffic coexistence in shared links.

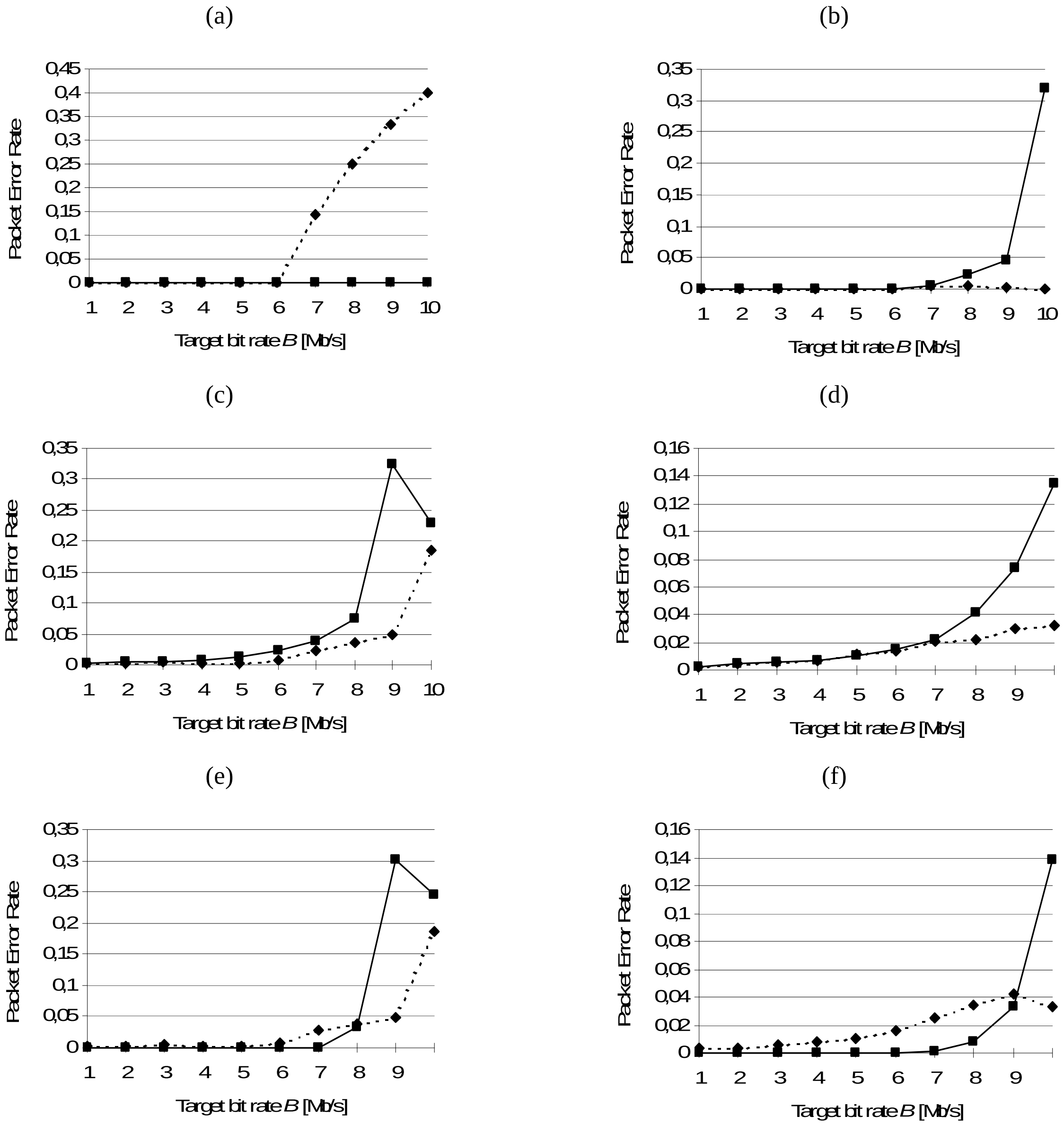
In both experiments, application of ECN signaling lightly improves performance of the TCP, but, generally has minor influence on coexistence of elastic and inelastic traffic.

The last, but not least, way to improve performance of elastic and inelastic traffic in shared link, arises from the observation, that burstless traffic improves (or, a least, not worsens) protocols performance. This improvement occurs for both types investigated queues (RED and tail drop) and is more obvious in he case of inelastic traffic. Presented observations were used to build a new kind of mechanism – the burst control mechanism – of the RTP transport protocol. Analysis of burs control was presented in the paper [7].

V. CONCLUSION

When different types of traffic (e.g. TCP and RTP) are sent in shared link it can result in one type of traffic having negative effect on the other. This issue is especially up-to-date since the percentage of the RTP traffic sent in the Internet increases.

In this paper we have presented simulation results showing the effect met by one type of traffic when sharing the link with different type of traffic. Simulation experiments were run using different queue management algorithms. Simulation results show that in a network, which is well dimensioned for multimedia, RTP isn’t a killing protocol for TCP. In such a network, we can observe acceptable RTP transmission (both packet error rate is smaller than an unacceptable value of 5% and transmission meet real-time conditions), as well as



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 5. Packet error rate of RTP flow (dotted line) and packet error rate of TCP flow (solid line) as a function of target bit rate *B* of CBR stream: a,b) DT experiment, c,d) RED experiment, e,f) RED&ECN experiment, a,c,e) bursty traffic, b,d,f) burstless traffic.

acceptable TCP transmission. Moreover, debursting of RTP traffic results in increasing of this tendency.

REFERENCES

[1] M. Freire, M Pereira. (eds.), "Encyclopedia of Internet technologies and applications", Information Science Reference, 2008.

[2] S. Floyd, M. Handley, J. Padhye, "TCP Friendly Rate Control (TFRC): protocol specification" *Internet-Draft draft-ietf-dccprfc3448bis-03.txt*, Nov. 2007.

[3] D. Chiu, A. Tam, „Network Fairness for heterogonous applications", in *Proc. ACM Sigcomm Asia Workshop*, Beijing, Apr. 2005.

[4] A. Tam, D. Chiu, J. Lui, Y. Tay, „A case for TCP-Friendly admission Ccontrol", in *Proc. IEEE International Workshop on Quality of Service*, pp. 229-238, June 2006.

[5] T. Bu, J. Liu, D. Towsley, „On the TCP-Friendliness of VoIP traffic", in *Proc. INFOCOM 2006*, pp. 1-12, Apr. 2006.

[6] http://nsnam.isi.edu/nsnam/index.php/Main_Page

[7] A. Chodorek, R. Chodorek, "An analysis of TCP-tolerant real-time multimedia distribution", in *Proc. HETNETs 2008*, Karlskrona, Sweden, February 2008.