Priority Forcing Scheme: A New Strategy for Getting Better than Best Effort Service in IP-based Network

Wojciech Burakowski and Monika Fudała
Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland.
wojtek@tele.pw.edu.pl, mkrol@tele.pw.edu.pl

Abstract: The paper describes a new strategy for source traffic generating, named Priority Forcing Scheme (PFS), that can be attractive for Internet users to transfer its data traffic with better quality than this available in current IP-based network, in which best-effort service is the only one. The PFS scheme assumes that an application, called PFS application, sends to the network a volume of additional traffic for the purpose of making the reservations for the data traffic in the overloaded router queues along the packet path in the network. The emitted redundant packets, named R-packets, should be rather of small size comparing to the data packets, named D-packets. The PFS scheme assumes that the R-packets waiting in a queue can be replaced by the arriving D-packets and belonging to the same flow. In this way, the D-packets can experience a prioritised service comparing to the packets produced by a non-PFS application. Notice that the proposed solution does not require any QoS (Quality of Service) mechanisms implemented in the network, like scheduler, dropping, marking etc., except R and D packets identification and replacing. The paper discusses different strategies for sending additional R-packets. Included simulation results show that traffic emitted by a PFS application can get better than best effort service, in limit case handled like the highest priority traffic. Depending on the volume of redundant traffic, a different level of prioritised service can be obtained.

Key words: IP-based network, better than best effort service, PFS – Priority Forcing Scheme

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35616-7_23

W. Cellary et al. (eds.), *Internet Technologies, Applications and Societal Impact*

1. INTRODUCTION

At present, the Internet users who want to get faster transfer of its data have no any additional mechanisms for doing it, even if it could be associated with an additional charging. This is due to the best effort service, the only one supported by current IP-based networks. On the other hand, now a number of attractive Internet applications is available, like VoIP, NetMeeting etc., but they are rather rarely used by a user since they require better service than this offered by best effort. More specifically, lower packet delay and lower packet losses are needed to satisfy the user. As a consequence, usefulness of these applications is limited, e.g. can be used during the time when Internet is under-loaded.

The paper addresses to the problem of providing better than best effort service in the IP network. One can distinguish between two main directions for doing it. The first approach is aimed at providing some QoS (Quality of Service) guarantees into the Internet, as e.g. in ATM (Asynchronous Transfer Mode). This approach assumes the IP QoS network concept, which can be based on an enhancement of DiffServ [3,4] or IntServ [5] architecture. However, this requires implementation of new QoS mechanisms at both the packet (e.g. conditioning, scheduling) as well as the network level (e.g. admission control, bandwidth broker). The example of new IP QoS architecture, based on DiffServ, is e.g. the AQUILA concept [6,7]. The second investigated way is to assure for selected flows better than best effort service. The simplest approach for doing it is the implementation of PQ (Priority Queuing) scheduling mechanism in IP routers [1]. However, this mechanism offers much better service for high priority traffic but may cause significant service degradation of lower priority traffic during time the router is in congestion. Another commonly used scheduling mechanism is WFQ (Weighted Fair Queuing) [2], which gives a possibility for a number of flows to get access to the link capacity proportionally to the a priori assigned weights. This allows us to get a fairness in the access to the link capacity between submitted flows. Other investigated way for achieving better than best effort service is to implement additional traffic control mechanisms at the application level. The example is some audio and video applications with quality adaptation mechanisms used to deal with end-to-end loss and delay variation [8]. Another proposal, named ABE (Alternative Best Effort), involving both application and network layer, is described in [9]. With ABE each packet is marked by application as green or blue. In the routers the green packets are handled only when their waiting times do not exceed the predefined delay. On the contrary, the service of blue packets is kept on the level offered by best effort service. In [10], a multi-level priority scheme is presented in which arriving packet from a prioritised flow reserves a place in the queue for the next packet belonging to this flow. In this mechanism, assigned priority is only kept when the rate of prioritised flows is relatively low. Thanks to that, the service degradation of lower priority packets does not take place, as happens in case of PQ system.

This paper describes a proposal for new mechanism allowing an Internet user to achieve better than best effort service [9]. This mechanism is named Priority Forcing Scheme (PFS) and can support an application to force priority service for the generated traffic in non-priority best-effort IP network. The PFS scheme assumes that the application, called PFS application, sends to the network a volume of additional traffic for the purpose of making the reservations for the data traffic in the overloaded router queues along the packet path in the network. The emitted redundant packets, named R-packets, should be rather of small size comparing to the data packets, named D-packets. According to PFS, the waiting in a queue R-packets can be replaced by the arriving D-packets belonging to the same flow. In this way, the D-packets can experience a prioritised service comparing to the packets produced by a non-PFS application. An advantage of the proposed solution is such that it does not require any QoS mechanisms implemented in the network like scheduler, dropping, marking etc., except R- and D-packets identification and replacing. The priority gained by a PFS application is relative and depends on the volume of sending reservation traffic as well as on the total volume of traffic submitted by other PFS (and partially, by non-PFS) applications.

The rest of the paper is organised as follows. Section 2 describes the proposed FPS mechanism in details. Evaluation of the effectiveness of the proposed mechanism in forcing priority in non-QoS IP networks is illustrated by numerical examples presented in section 3. Section 4 shows the advantages of using PFS mechanism in the case of VoIP application. Section 5 introduces us to the implementation issues of the PFS mechanism. Finally, section 4 summarises the paper.

2. PFS MECHANISM

The PFS mechanism is a proposal for allowing the Interenet users to get better than best effort service in non-QoS IP network. To data traffic generated by the PFS application a volume of additional traffic is added. As depicted in *Figure 1*, the stream of packets generated by a PFS application consists of data packets, D-packets, and reservation packets, R-packets. The aim of R-packets is to make the reservations in the queue for D-packets.

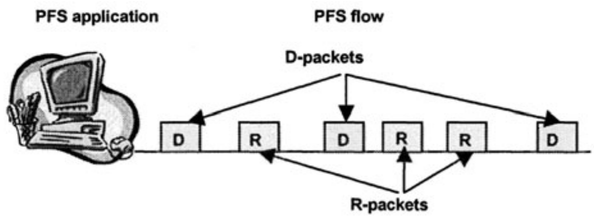

Figure 1. Packets generated by PFS application (PFS flows): data (D-packets) and redundant packets (R-packets)

The way of handling D- and R-packets in a router output port, represented in this case by single server queuing system with finite waiting room is depicted in

Figure 2. When R-packet arrives, it will enter the system only if the server is idle or there is free room in the queue, otherwise it will be dropped. When D-packet is arriving, first of all it is checked if any R-packet from the same flow is currently waiting. If such situation takes place then D-packet replaces the first from the top R-packet (this R-packet is dropped). Otherwise, the D-packet is served in best effort way and this means that it occupies the last position in the current queue or is lost when no waiting room is available.

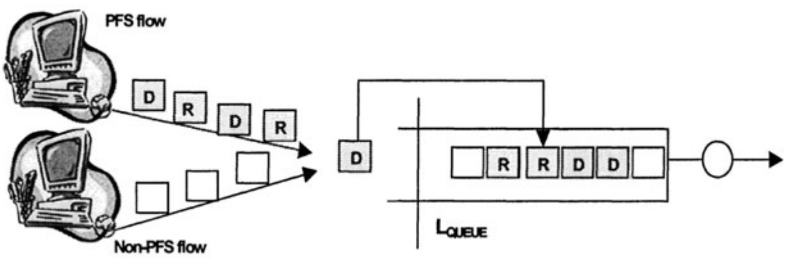

Figure 2. The way of handling D- and R-packets in the router output port

3. EVALUATION OF PFS MECHANISM

This section contains the simulation results allowing us for evaluation of the PFS scheme. Main attention will be focused on the following:

- Strategy for generating R-packets;
- Gained priority level of the PFS traffic under different traffic conditions (including overload conditions);
- PFS mechanism with multi-level priorities.

3.1 Strategy for generating R-packets

Now, we focus on a strategy for generating R-packets. One can expect that by sending more R-packets the higher priority may be reached. However, a user should be aware that:

- The gained priority level for D-packet traffic is of relative type and depends on cumulative traffic generated by other running PFS applications;
- Additional traffic related to R-packets is charged, even if the R-packets are as short as possible.

In this section the results corresponding to two strategies for generating R-packets by PFS application are reported in the case the traffic is submitted to the single server queue with infinite waiting room (as depicted in *Figure 3*). The system is fed by two packet flows, each of Poissonian type and negative exponential packet length:

- Foreground flow (PFS or non-PFS flow) with load ρ_{FOR_PFS} = 0,1 and mean service rate μ_{FOR_PFS} = 1 (the load caused by D-packets is negligible);
- Background flow (non-PFS flow) with load ρ_{BACK} = 0,85 and mean service rate μ_{BACK} = 1;

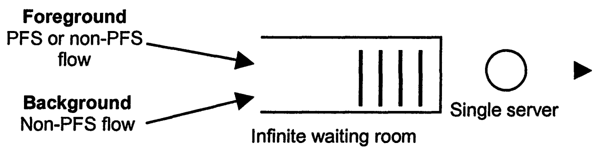

Figure 3. Single server queue with infinite waiting room fed by PFS and non-PFS traffic

In the first case, the reference one, we assume that the knowledge of D-packets inter-arrival times in PFS flow is known by the application, and thanks to that we can generate a number of R-packets between two consecutive D-packets in a control way. The best results, from the point of view of D-packet delay characteristics, were obtained when R-packets were uniformly distributed between two consecutive D-packets. *Table 1* shows the results corresponding to mean waiting times for both the foreground D-packets and background packets as well as the probability that D-packet replaces R-packet (P_rep) vs. number of R-packets between two consecutive D-packets. Additionally, the last row in the table includes results obtained for PQ system, where foreground and background flows (both non-PFS flows) were handled in separate queues with assigned higher priority for foreground flow.

Table 1. The impact of the number of R-packets sent between two consecutive D-packets in PFS flow (D-packet inter-arrival time is known). E_Wq – the mean waiting time for D-packets/background packets, P_rep – the probability that D-packet replaces R-packet

# of R-packets sent between consecutive D-packets in foreground PFS flow	Foreground $\rho_{FOR} = 0,1$		Background $\rho_{BACK} = 0,85$
	E_Wq	P_rep	E_Wq
0 (FIFO)	19,0		19,0
1	16,4	0,78	19,3
2	6,2	0,82	20,4
5	3,2	0,9	21
9	2,2	0,91	21
19	1,6	0,93	21,1
79	1,2	0,94	21,4
0 (PQ)	1,05		21,1

In the case 2 we assume that inter-arrival times in the D-packet stream are unknown and, therefore, the R-packets are generated with constant rate in this time. The obtained results are included in *Table 2.*

Table 2. The impact of PFS_ratio (packet interarrival time is unknown, R-packets are generated with constant rate after each D-packet). E_Wq – mean waiting time for D-packets/background packets, P_rep – the probability that D-packet replaces R-packet, PFS_ratio is the ratio of mean rate of R-packets to mean rate of D-packets

PFS_ratio	Foreground $\rho_{FOR} = 0,1$		Background $\rho_{BACK} = 0.85$
	E_Wq	P_rep	E_Wq
0 (FIFO)	19,0		19,0
1	14,6	0,52	19,6
2	8,2	0,72	20,3
5	4,6	0,83	20,7
9	3,6	0,86	20,7
19	2,8	0,87	20,9
79	1,9	0,92	21
0 (PQ)	1,05		21,1

One can summarise the results from Table 1 and 2 as follows:

- PFS mechanism enables applications for forcing priority up to the pure priority (as in ordinary PQ scheme);
- The strategies for generating R-packets in the considered cases give similar results;

– The gained priority level depends on the volume of sending R-packets. By increasing PFS_ratio the probability that D-packet replaces R-packet is also increased and, as a consequence, higher priority can be achieved.

3.2 Gained priority vs. volume of PFS traffic

In this section we present results showing the relations between gained priority for packets from given PFS application and volume of total submitted PFS traffic. For this purpose, we now consider single server queue with two options: finite and infinite queue size. Once again, two packet flows, each of Poissonian type and negative exponential packet lengths, are submitted to the system. They are:

– Foreground flow (PFS flow or non-PFS flow) with load $\rho_{FOR} = 0{,}1 \div 0{,}95$ and mean service rate $\mu_{FOR} = 1$ (the load of R- packets is negligible);
– Background flow (non-PFS flow) with load $\rho_{BACK} = 0{,}85 \div 0$ and mean service rate $\mu_{BACK} = 1$.

The reported below results were obtained assuming that the total system load is constant and equal to 0,95 (heavy load conditions). Considered traffic cases are presented in Table 3.

Table 3. Considered traffic cases

	Foreground (FOR)	**Background (BACK)**
Case #1 (FIFO)	Non-PFS flow	Non-PFS flow
Case #2	PFS flow, PFS_ratio = 1	Non-PFS flow
Case #3	PFS flow, PFS_ratio = 2	Non-PFS flow
Case #4	PFS flow, PFS_ratio = 3	Non-PFS flow
Case #5 (PQ)	Non-PFS flow	Non-PFS flow

Figure 4 shows the values of mean waiting times (E_Wq) for foreground (excluding R-packets) and background flows as a function of system load corresponding to foreground flow (ρ_{FOR}). Results were collected assuming infinite waiting room with total load equals 0.95. These results say that despite the amount of PFS traffic is increasing, the PFS mechanism still enables application for forcing priority. Therefore, one can conclude that the amount of PFS traffic associated to given application has no essential impact on the priority level forced by this application. Furthermore, by increasing PFS_ratio, the gained priority is tending to priority adequate for PQ scheme.

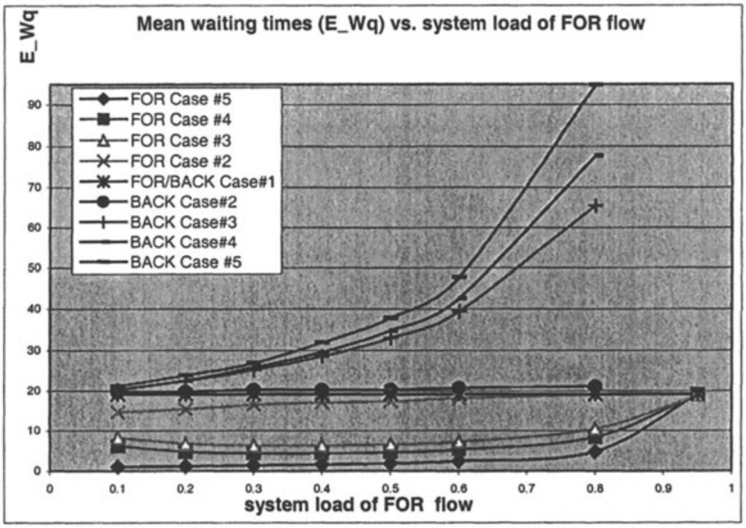

Figure 4. Infinite waiting room case with total load = 0.95: mean waiting times (E_Wq) (excluding R-packets in PFS flow) vs. system load of foreground flow (ρ_{FOR}).

Figure 5 shows probability of packet losses (P_loss) for foreground (excluding R-packets) and background flows as a function of system load of foreground flow (ρ_{FOR}), still keeping total load equals 0.95. In this case, the considered system was with finite waiting room, L_{QUEUE}=10. As it was expected, one can observe packet losses also from PFS flow. However, PFS applications is still able to force priority up to the level available in PQ scheme.

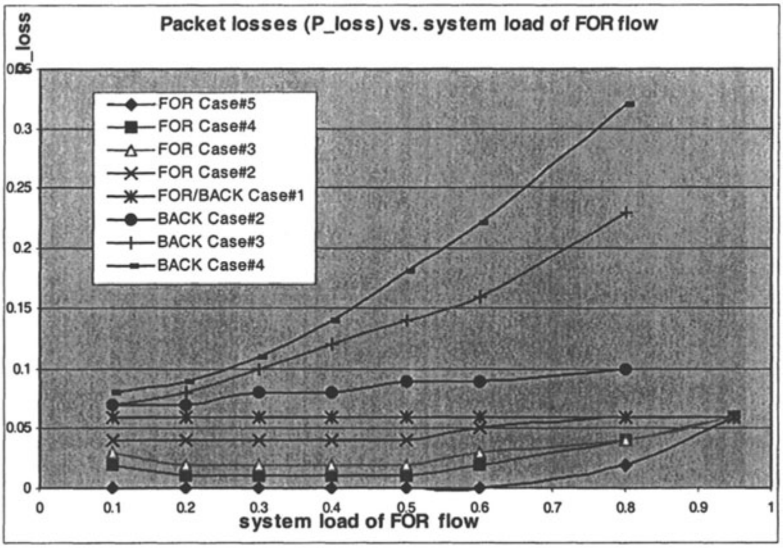

Figure 5. Finite waiting room (L_{QUEUE}=10 packets) case with total load=0.95: probability of packet losses (P_loss) (excluding R-packets in PFS flow) vs. system load of foreground flow (ρ_{FOR})

3.3 Efficiency of PFS mechanism in overload conditions

In this section we show the results illustrating efficiency of the PFS mechanisms in the case, the system is overloaded (traffic submitted to the system exceeds the system capacity). The considered single queue system is with finite waiting room (L_{QUEUE} = 10). The system is fed by two packet flows, both of Poissonian type and negative exponential packet length:

- Foreground flow (PFS flow or non-PFS flow) with load ρ_{FOR} = 0,2 and mean service time μ_{FOR} = 1 (the load of R-packets is negligible);
- Background flow (non-PFS flow) with load ρ_{BACK} = 0,8 ÷ 0,95 and mean service time μ_{BACK} = 1;

The offered traffic to the system, ρ_{SYS}, is now increasing from 1 to 1,15. Considered traffic cases are the same as in *Table 3*. *Figure 6* shows packet loss probability (P_loss) for foreground (excluding R-packets) and background flows when the system is overloaded by background traffic. Therefore, one can conclude that PFS flow is still able to force priority even if the system is overloaded. However, the gained priority level decreases when overload of the system increases.

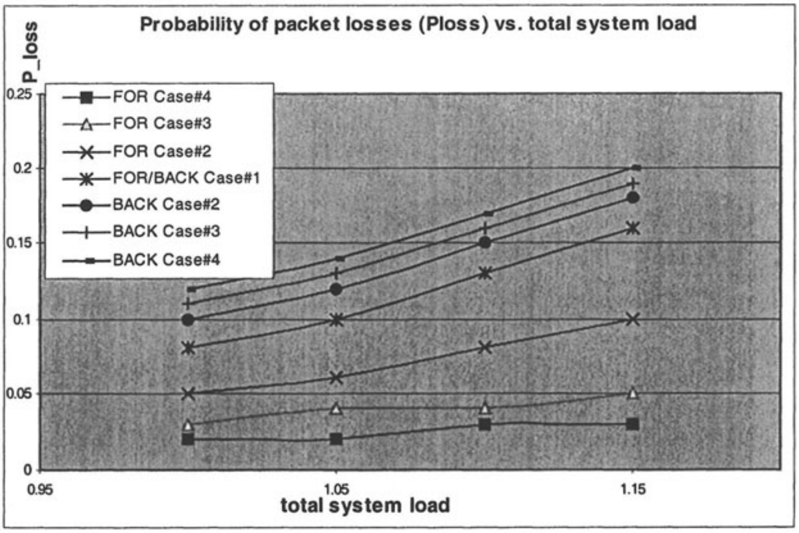

Figure 6. Overloaded conditions in the single server system with queue of 10 packets: probability of packet losses (Ploss) (excluding R-packets in PFS flow) vs. system load (r_{SYS}).

3.4 Multi-level priority scheme based on PFS mechanism

In this section we will focus on efficiency of a multi-level priority scheme based on PFS mechanism. Such scheme is possible to get since the gained priority level depends on assumed volume for redundant R-packets. The studied now system is depicted in *Figure 7* and it consists of single server with infinite waiting room. The system is fed by a number of packet flows, both of Poissonian type and negative exponential packet length:

- Foreground flows (PFS flows #1,#2 and#3), mean service rate $\mu_{FOR} = 1$ (the load of R-packets is negligible);
- Background flow (non-PFS flow) – mean service rate $\mu_{BACK} = 1$;
  The total system load is constant and equal to 0,95.

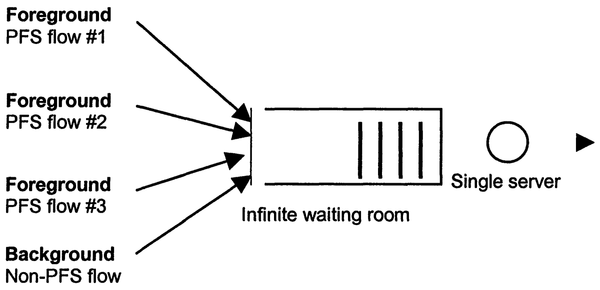

Figure 7. The studied system.

Considered traffic conditions are presented in *Table 4*.

Table 4. Considered traffic cases

Foreground PFS flow #1		Foreground PFS flow #2		Foreground PFS flow #3		Background
$\rho_{FOR\ PFS1}$	PFS_ratio	$\rho_{FOR\ PFS2}$	PFS_ratio	$\rho_{FOR\ PFS3}$	PFS_ratio	ρ_{BACK}
0,2	2	0,2	3	0,2	$4 \div 79$	0,35

Table 5 shows the results corresponding to mean waiting times and probabilities that D-packet replaces R-packet for particular PFS flows assuming that the volume of redundant traffic for flows#1 and #2 is fixed (PFS_ratio is equal to 2 and 3, respectively) while for flow#3 is changed from 4 up to 79.

Table 5. The results for three priority levels

PFS_ratio for PFS flow #3	Foreground PFS flow #1 PFS_ratio = 2		Foreground PFS flow #2 PFS_ratio = 3		Foreground PFS flow #3 PFS_ratio = 4÷ 79		Background
	E_Wq	P_ rep	E_Wq	P_ rep	E_Wq	P_ rep	E_Wq
4	10,3	0,84	6,2	0,87	4,8	0,89	39,3
39	10,9	0,85	6,7	0,88	2	0,93	40,5
79	11	0,85	6,8	0,88	1,8	0,94	40,8

Therefore, one can conclude as follows:

- The priority level gained by PFS application is of relative type and, in addition it slightly depends on cumulative traffic generated by other running PFS applications (number of running PFS applications and their PFS_ratios);
- The flow with the highest PFS ratio has higher priority comparing to other PFS flows.

4. USING PFS MECHANISM FOR VOIP APPLICATION

Now, we show the usefulness of using PFS mechanism for getting better quality in the case of VoIP application, which is extremely desirable. The VoIP is sending traffic with constant bit rate equals 64 kbps and packets of 100 bytes. This traffic is submitted to the network with 3 routers, as it is depicted in Figure 8. The inter-router links, C_1 and C_2, are of 2 Mbps each. The buffer size at the output router port is fixed to 40 packets.

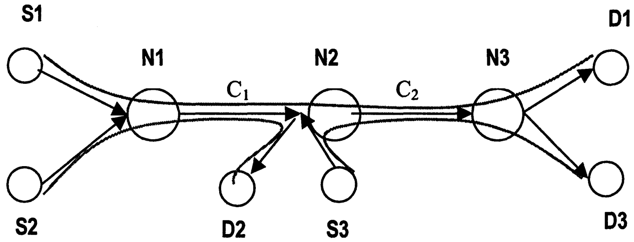

Figure 8. Network topology

The foreground connection for VoIP is established between S1-D1 end users and passes the routers N1, N2 and N3. The background traffic, of Poissonian type, is produced by non-PFS applications and is carried between S2-D2 and S3-D3. For this traffic the size of the packets is done by exponential distribution with the mean 750 bytes. The traffic conditions are the following:

- Case 1: C_1 and C_2 are both on the heavy load conditions (ρ= 0.95);
- Case 2: C_1 is under heavy load conditions (ρ= 0.95), while C_2 is overloaded (ρ=1.1);
- Case 3: C_1 is overloaded (ρ=1.1), while C_2 is under heavy load conditions (ρ= 0.95).

Let us recall that for transferring voice on acceptable level, the requirements are: for one-way delay – not more than 150 ms, for packet loss – less than 10^{-4}. *Table 6,* 7 , 8 show the received results illustrating the quality of handling PFS and non-PFS traffic, corresponding to the case 1, 2 and 3 respectively. In this scenario we have assumed that PFS_ratio is equal to 3.

Table 6. Case 1: The Packet transfer characteristics measured on C_1 and C_2 links. E_Lq (Max_Lq) – the mean (maximal) queue size observed by D-packets/background packets, E_Wq (Max_Wq) - the mean (maximal) waiting time for D-packets/background packets, P_rep – the probability that D-packet replaces R-packet, P_loss - probability of packet losses for foreground (excluding R-packets) and background flows. (*) The percentage of D-packets which experience end-to-end delay above 150 ms is 10^{-5}

	Foreground flow (PFS_ratio = 3)				**Background flow**		
	E_Lq/ Max_Lq [in packets]	**E_Wq/ Max_Wq** [ms]	**P_rep**	**P_loss**	**E_Lq/ Max_Lq** [in packets]	**E_Wq/ Max_Wq** [ms]	**P_loss**
C_1 link	1,6/21	4,8/86,6	0,88	-	15,3/40	28,4/172,9	$1{,}9*10^{-2}$
C_2 link	4,2/40	11,7/193,5	0,69	$3*10^{-5}$	14,5/40	30,7/193,8	$1{,}6*10^{-2}$
	Mean end-to-end delay		**Max end-to-end delay**				
	17,7		(*) 201,0				

Table 7. Case 2: The Packet transfer characteristics measured on C_1 and C_2 links. The percentage of D-packets which experience end-to-end delay above 150 ms is 10^{-4}

	Foreground flow (PFS_ratio = 3)				**Background flow**		
	E_Lq/ Max_Lq [in packets]	**E_Wq/ Max_Wq** [ms]	**P_rep**	**P_loss**	**E_Lq/ Max_Lq** [in packets]	**E_Wq/ Max_Wq** [ms]	**P_loss**
C_1 link	1,6/21	4,8/86,6	0,88	-	15,3/40	28,4/172,9	$1{,}9*10^{-2}$
C_2 link	7,4/40	19,2/196,5	0,93	$6{,}3*10^{-4}$	28,5/40	60,8/207,0	$9{,}9*10^{-2}$
	Mean end-to-end delay		**Max end-to-end delay**				
	25,2		(*) 210,7				

Table 8. Case 3: The Packet transfer characteristics measured on C_1 and C_2. The percentage of D-packets which experience end-to-end delay above 150 ms is $3*10^{-5}$

	Foreground flow (PFS_ratio = 3)				**Background flow**		
	E_Lq/ Max_Lq [in packets]	**E_Wq/ Max_Wq** [ms]	**P_rep**	**P_loss**	**E_Lq/ Max_Lq** [in packets]	**E_Wq/ Max_Wq** [ms]	**P_loss**
C_1 link	2,0/23	5,7/90,3	0,98	-	28,5/40	52,4/186,0	10^{-1}
C_2 link	5,1/40	14,0/184,5	0,65	10^{-4}	14,3/40	31,7/195,4	$1{,}5*10^{-2}$
	Mean end-to-end delay		**Max end-to-end delay**				
	20,9		(*) 231,5				

The presented results say that the transfer quality of traffic associated to considered VoIP application using PFS mechanism is much better (about 3-4 times) than it could be experienced without this mechanism. Better results can be obtained by increasing PFS_ratio.

5. IMPLEMENTATION ISSUES OF PFS MECHANISMS

The implementation of PFS mechanism requires: (1) from application - a possibility for sending additional traffic in control way and dropping this traffic at the end-point, (2) from router – the mechanism for recognition D- and R-packets, and possibility for replacing R- by D-packets. More detailed studies are needed for evaluating the complexity of the proposed solutions. For now, we believe that it is possible to do them and it seems that implementation of these mechanisms at application as well as at router level is not so complex.

6. SUMMARY

In the paper the PFS mechanism for forcing priority handling in IP-based network was described and its effectiveness was evaluated from different points of view. This mechanism, implemented at the application layer, assumes sending additional R-packets for reservation purposes of D-packets. The presented numerical results illustrate the potential benefits of using the PFS mechanism. In particular, the VoIP application can transfer its traffic by overloaded best effort IP-based network with quality still acceptable for the users.

Further studies are required with main focus on practical verification of the proposed PFS mechanism. On the other hand, the strategies for sending R-packets in the case of elastic traffic, like TCP-controlled traffic, is also of interest.

REFERENCE

[1] S.Keshav, *An engineering approach to computer networking*, chapter 9: Scheduling, Addison-Wesley, 1997

[2] Final report COST 242 (J.Roberts, U. Mocci, J. Virtamo eds.), *Broadband network teletraffic: Performance evaluation and design of broadband multiservice networks*, Lectures Notes in Computer Science 1155, Springer 1996.

[3] S. Blake et al., *An Architecture for Differentiated Services*, Internet RFC 2475, December 1998.

[4] Y. Bernet et al., *Differentiated Services, Internet Draft*, draft-ietf-diffserv-framework-0.2.txt, February 1999.

[5] R. Braden, D. Clark, and S. Shenker, *Integrated services in the Internet architecture: an overview*, RFC1633, June 1994.

[6] Deliverable D1301, *Specification of traffic handling for the first trial*, AQUILA project consorcium, http://www-st.inf.tu-dresden.de/aquil/, September 2000.

[7] Deliverable D3201, *First trial report, AQUILA project consorcium*, http://www-st.inf.tu-dresden.de/aquil/, March 2001.

[8] D. Wu et all., Streaming Video over the Internet: Approaches and Directions, IEEE Transaction on circuits and systems for video technology, March 2001.

[9] P. Hurley et all., *Providing a low delay service within best effort*, IEEE Network, May/June 2001.

[10] W. Burakowski, H.Tarasiuk, *New Priority Scheme for Differentiating QoS in IP Networks*, Proceedings of the 8th Polish Teletraffic Symposium, Zakopane, September 2001.