Paper

PFS scheme for forcing better service in best effort IP network

Monika Fudała and Wojciech Burakowski

Abstract—The paper presents recent results corresponding to a new strategy for source traffic generating, named priority forcing scheme (PFS), allowing Internet users for getting better than best effort service in IP network. The concept of PFS 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 IP 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 quality of service (QoS) mechanisms implemented in the network, like scheduler, dropping, marking etc., except R- and D-packets identification and replacing. We discuss the PFS efficiency for forcing priority in the overloaded conditions. Moreover simple system analysis is also presented. Finally, the profits of using PFS scheme are illustrated by examples corresponding to FTP (TCP controlled traffic) and VoIP (UDP streaming traffic) applications.

Keywords—IP-based network, better than best effort service, priority forcing scheme.

1. Introduction

At present, the Internet users who want to get faster transfer of their data have no 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. As a consequence, e.g., a file transfer protocol (FTP) user has to accept long upload/download file time when the network is overloaded. On the other hand, several attractive Internet applications are available now, like voice over IP (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.

One may observe two main areas of activities for introducing QoS into Internet. The first direction is aimed at providing some QoS guarantees, similarly as it was done for ATM. In this spirit, the IP QoS network concept is investigated, 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 [1, 2]. The second investigated direction is to assure for selected flows better than best effort service. The simplest approach for doing it is the implementation of priority queuing (PQ) scheduling mechanism [10] in IP routers. 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 weighted fair queuing (WFQ) [7, 10], which gives a possibility for a number of flows to get access to the link capacity proportionally to the a priori assigned weights. Other investigated way for achieving better than best effort service is to implement additional traffic control mechanisms at the application level. An example is some audio and video applications with quality adaptation mechanisms used to deal with end-to-end loss and delay variation [11]. Another proposal, named alternative best effort (ABE), involving both application and network layer, is described in [9].

The paper addresses to the strategy, named priority forcing scheme, introduced in [6]. The PFS is a proposal for achieving better than best effort service in the IP network, as it is defined, e.g., in [3]. The PFS mechanism can support an application to force prioritised packet service in IP best effort network. It 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 R-packets waiting in a queue can be replaced by the arriving D-packets belonging to the same flow. In this way, the D-packets could experience a prioritised service comparing to the packets produced by a non-PFS application. An advantage of the proposed solution is that any QoS mechanisms are implemented in the network, like scheduler, dropping, marking, etc., except R- and D-packets identification and replacing. As it was shown in [6], by using PFS a relative priority level can be reached. This paper includes recent results concerning PFS, and discusses the PFS efficiency for forcing priority in the overloaded conditions, as well as presents simple system analysis. Moreover, the profits of using PFS scheme are illustrated by considering examples corresponding to FTP (TCP controlled traffic) and VoIP (UDP streaming traffic) applications.

The rest of the paper is organised as follows. Section 2 gives short overview of PFS scheme. Section 3 presents simple system analysis. The capability of PFS for reducing packet waiting times in the case of overload conditions are discussed in Section 4. Profit from using PFS for getting better service by VoIP and FTP applications is illustrated in Section 5. Finally, Section 6 summarises the paper.

2. Overview of PFS mechanism

The PFS mechanism is designed to forcing prioritised service by a user, who wants to get better service in best effort network. It assumes that the user application besides the data packets, say D-packets, may also generate in a control way some additional packets, say R-packets, as depicted in Fig. 1. The R-packets are only generated for making the potential reservations for D-packets in the overloaded router queues. To minimise this redundant traffic in the network, which is extremely required to reduce additional load (and charging), the size of R-packets should be set as small as possible, i.e., 40 bytes for TCP and 28 bytes for UDP.

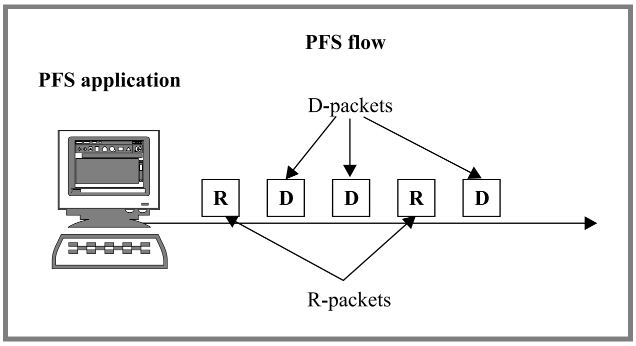

Fig. 1. Packet stream generated by PFS application (PFS flows): data, D-packets, and reservation packets, R-packets.

Since the considered network is with the only single class service, all packets in the router are served according to the FIFO discipline if no additional mechanisms exist. However in the PFS, the D- and R-packets are treated in different way (Fig. 2). For the R-packets the best effort service is assumed with a possibility of dropping them from the queue when a new D-packet arrives. For this D-packet the system is searching for the R-packet waiting in the queue (and belonging to the same PFS flow), which is the first from the top. If no R-packets exist, the D-packet is served according to the FIFO. If at least one R-packet is in the queue, the D-packet drops the R-packet and sizes its position. As a consequence, the D-packets are entitled to get better than best effort service when R-packets exist in the queue. Remark that D-packets are lost only if no R-packets exist in the queue and queue is full. One can expect that D-packets may get greater profit from PFS when more R-packets are generated to the network. Remark also that in the case of non-overloaded queue the service of D-packets is without any delay, as well as the R-packets are not dropped and are transmitted to the next router according to the routing rules. Then, the R-packet can be replaced by a D-packet only in the overloaded routers. Finally, the PFS can be effective in the situations when a bottleneck could occur at any router along the path.

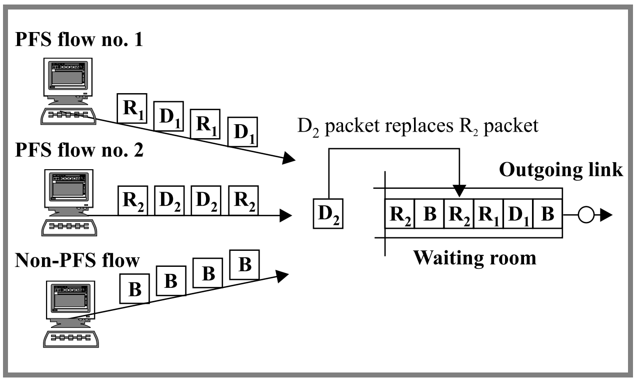

Fig. 2. Queue management for PFS mechanism: example illustrating rules for replacing R-packets in the queue by arriving D-packet.

Notice however, that implementation of PFS mechanism requires the following: (1) from application—a possibility for sending additional packets in a control way and dropping these packets (if any) at the ending-point, (2) from routers—the mechanism for distinguishing between D- and R-packets, and capabilities for replacing R- by D-packets.

3. Simple system analysis

In this section we present simple analysis of system using PFS scheme. Let us assume that the system (Fig. 3) is a single server with infinite waiting room and is fed by three types of flows, which are:

- Flow no. 1, which represents the D-packet flow emitted by a single PFS application. It is assumed as Poissonian stream with the rate λ_D and service times described by the negative exponential distribution with parameter μ_D.
- Flow no. 2, which represents the R-packet flow emitted by the PFS application generating flow no. 1. The R-packets are emitted periodically, at each T_R interval. Furthermore, let us assume that the load of this flow is negligible (R-packet size is close to 0). As it was shown in [6], sending R-packets with constant rate is the simplest and effective way for getting a profit from PFS.
- Flow no. 3, which represents the cumulative flow emitted by other sources (supported by PFS and non-supported by PFS). All B-packets are served by the system in best effort way. We assume that B-packets arrive accordingly to Poissonian low with the rate λ_B and service times described by negative exponential distribution with parameter μ_B.

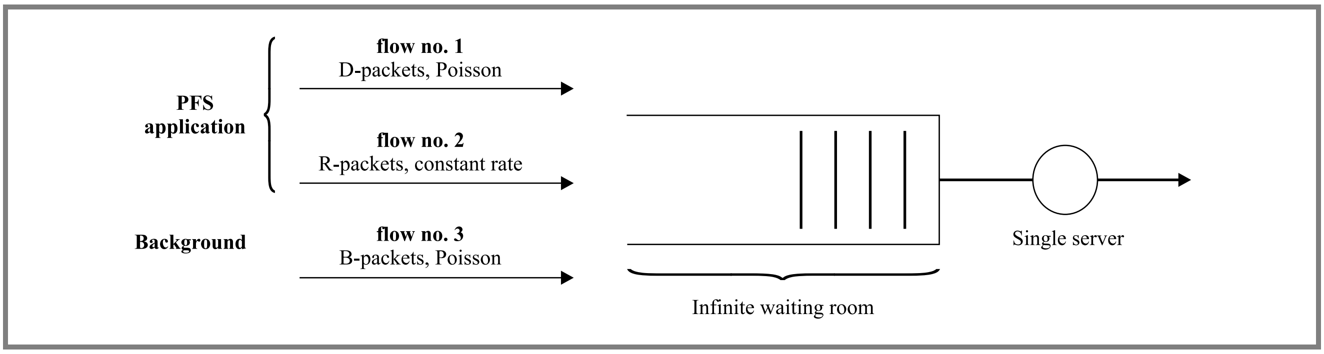

Fig. 3. Single server queue with infinite waiting room fed by PFS and non-PFS traffic.

Assuming that $\mu_D = \mu_B = \mu$, the considered system is similar to the M/M/1 queue, with the only difference that now arriving D-packet may size the R-packet in the queue (if any), and in this way get better service.

Now, we use the expression from M/M/1 system analysis, determining distribution of the packet waiting times $W_q(T)$ (e.g., [8]), which is:

$$W_q(T) = \Pr(t \le T) = 1 - \rho \cdot e^{-\mu(1-\rho)T}, \qquad (1)$$

where $\rho = (\lambda_D + \lambda_B)/\mu$ (remind that service times of R-packets are equal to 0).

Taking Eq. (1) and knowing that R-packets enter system at each T_R interval, we deduce the following approximate formula for probability that at the moment of D-packet arrival it "sees" n $(n = 0, 1, \ldots,)$ R-packets in the queue, assuming that R-packets are not replaced by D-packets:

$$\Pr{}_D^R(n) = \\ = \begin{cases} W_q(0.5\,T_R) & \text{for } n = 0 \\ W_q\left(T_R(n+0.5)\right) - W_q\left(T_R(n-0.5)\right) & \text{for } n = 1, 2, \ldots \end{cases} . \qquad (2)$$

Consider that a D-packet is entering the system at time t_0. Assuming that in this moment there are n $(n = 0, 1, 2, \ldots)$ R-packets in the queue, we deduce that the first from these R-packets arrived to the system at time $t_0 - \Delta t$, where $\Delta t = (0.5\,T_R + (n-1)T_R)$. However, during the interval Δt a number of D-packets could arrive to the system and replace R-packets. Probability that k $(k = 0, 1, 2 \ldots)$ D-packets arrived to the system during the interval Δt is done by:

$$\Pr(k, \Delta t) = \frac{(\lambda_D\,\Delta t)^k}{k!}\, e^{-\lambda_D \Delta t}. \qquad (3)$$

From Eqs. (2) and (3), we deduce approximate formula for average number of R-packets (not-replaced by D-packets) in the queue at the moment a D-packet enters the system, say N_R, which is:

$$N_R = \sum_n \sum_{i=0}^{i=n} i \Pr{}_D^R(n) \cdot \Pr\left(n - i, 0.5\,T_R + (n-1)T_R\right). \qquad (4)$$

Remark that Eq. (4) evaluates a lower bound of average number of R-packets in the queue. It can be explained in this way that in Eq. (4) we assumed that all D-packets in the queue have replaced R-packets. In fact, it is not truth since during T_R interval more than one D-packet may enter the system.

Finally, we introduce a measure allowing us to evaluate the profit coefficient (p_f) we could get from PFS. Remark that for the system without PFS, which is modelled in this case by M/M/1 system with FIFO discipline, the $p_f = 0$. The definition of the profit coefficient is as follows:

$$p_f = \lambda_B N_R T_R \,/\, \mu_B\,. \qquad (5)$$

Other interesting measure, illustrating the profit we could get from PFS, is the probability that a D-packet will replace R-packet, say p_s. The p_s denotes the percentage of D-packets handled in better than best effort way and it could be evaluated by:

$$p_s = 1 - W_q(0.5\,T_R)\,. \qquad (6)$$

4. Priority forcing scheme capability in the case of overloaded queue

In this section we show effectiveness of PFS scheme for forcing priority in the queue overloaded conditions. For this purpose we consider the system from Fig. 3. We expect, that according to definition (5), the profit an application can get from using PFS is greater when number of waiting packets is growing. The ideal PFS behavior will be if we are able to provide constant waiting times for D-packets, independently of volume of submitted background traffic. Anyway, one can expect that by increasing generating rate of R-packets the effectiveness of PFS is also increased.

In Fig. 4 are presented the results showing effectiveness of PFS for forcing priority as a function of number of D- and B-packets being in the queue (L_q), at the moment a D-packet arrives, assuming that $\mu_D = \mu_B = \mu = 1$, $\lambda_D = 0.1$, $\lambda_B = 0.85$. Notice, that when $L_q = 0$, any priority forcing mechanism is needed. The Fig. 4a corresponds to the case when distance between consecutive arriving R-packets, T_R, is equal to the mean interarrival time of D-packets, $1/\lambda_D$, while Fig. 4b corresponds to the case when $T_R = 1/(2 \cdot \lambda_D)$. Four characteristics are presented: *1*—mean number of R-packets being in the queue, *2*—reduced D-packet waiting times (number of waiting B-packets the D-packet "jumps over") thanks to PFS scheme, *3*—experienced mean D-packet waiting time using PFS, and *4*—mean D-packet waiting time for the system without PFS.

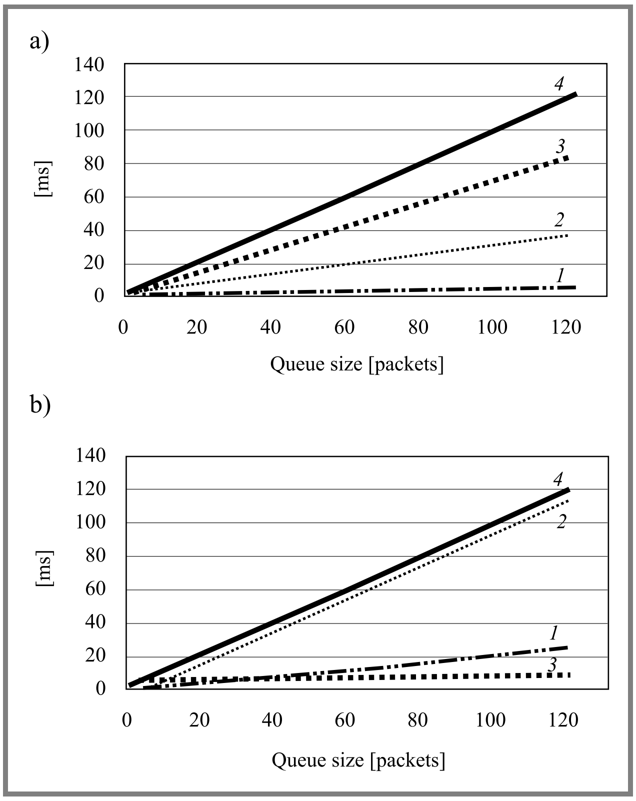

Fig. 4. Results showing effectiveness of PFS for forcing priority as a function of number of D- and B-packets being in the queue, at the moment a D-packet arrives, assuming that $\mu = \mu_D = \mu_B = 1$, $\lambda_D = 0.1$, $\lambda_B = 0.85$: (a) $T_R = 1/\lambda_D$; (b) $T_R = 1/(2 \cdot \lambda_D)$. Explanations: *1*—mean number of R-packets in the queue seen by D-packet; *2*—PFS: reduced D-packet waiting times; *3*—PFS: mean D-packet waiting times; *4*—non-PFS: mean D-packet waiting times.

The obtained results show that by applying PFS scheme one may get essential improvement of packet delay transfer characteristics comparing to the system without PFS. The observation is that the profit gained by using PFS increases when the number of waiting packets is growing. This profit depends on the rate the R-packets are generated. Notice, that in this way we may shape the waiting times for D-packets. In the presented experiment (Fig. 4b), the waiting times for D-packets are almost constant and low, independently on the temporary queue size. In this case number of generated R-packets is double (in the average sense) comparing to emitted D-packets. This result is very promising. It appears that by appropriate setting of PFS mechanism parameters we are able to get excellent packet transfer characteristic, as, e.g., desirable by VoIP application.

5. Applying PFS to VoIP and FTP

In this section we present the simulation results showing efficiency of using PFS mechanisms to improve delay packet transfer characteristics in the case of VoIP and FTP applications. As VoIP is typical for applications emitting streaming packet flows, the FTP is for file transfer and belongs to elastic applications with TCP-controlled packet sending rate depending on network conditions.

5.1. VoIP application

Now, we show the usefulness of using PFS mechanism for getting better quality by VoIP application. The tested VoIP is sending traffic with constant bit rate equals to 64 kbit/s and fixed packet size of 100 bytes. This traffic is submitted to the network with 3 routers, as depicted in Fig. 5. The inter-router links, N1↔N2 and N2↔N3 are of 2 Mbit/s each, the capacity of access links to the routers is 10 Mbit/s. The buffer size at the output router port is fixed to 40 packets.

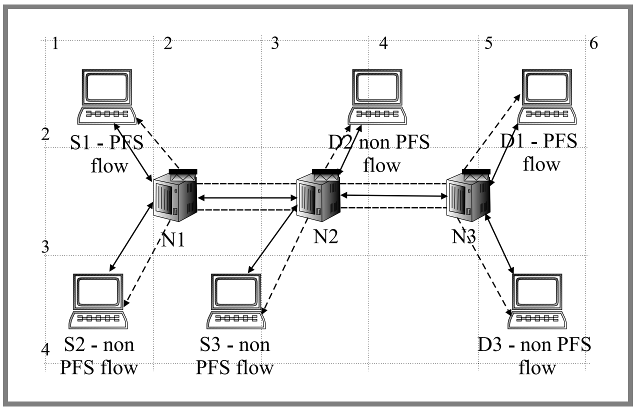

Fig. 5. Network topology for testing VoIP.

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 also constant and equals to 750 bytes. We consider three cases depending on traffic conditions in the tested network, which are:

Case 1. The links N1↔N2 and N2↔N3 are both on the heavy load conditions ($\rho = 0.95$).

Case 2. The link N1↔N2 is under heavy load conditions ($\rho = 0.95$), while the link N2↔N3 is overloaded ($\rho = 1.1$).

Case 3. The link N1↔N2 is overloaded ($\rho = 1.1$), while the link N2↔N3 is under heavy load conditions ($\rho = 0.95$).

Let us recall that for transferring voice on acceptable level, the requirements are: for one-way delay—not more

Table 1
End-to-end B- and D-packet transfer characteristics versus R-packet flow rate (V_R)

Generation rate	Node N1/N2	PFS flow S1-D1		Non-PFS flow S2-D2		Non-PFS flow S3-D3	
V_R [kbit/s]	p_s	$D_m/D_{\max}$ [ms]	p_{loss}	D_m [ms]	p_{loss}	D_m [ms]	p_{loss}
Case 1							
0	–	54.2/206.6	$2\cdot10^{-3}$	32.4	$1.7\cdot10^{-3}$	30.5	$1.1\cdot10^{-3}$
18	0.8/0.14	48.6/204.9	$9.8\cdot10^{-4}$	32.6	$1.9\cdot10^{-3}$	30.8	$1.2\cdot10^{-3}$
36	0.88/0.62	21.1/123.2	$4\cdot10^{-5}$	34.2	$4.2\cdot10^{-3}$	31.6	$1.8\cdot10^{-3}$
54	0.92/0.77	14.7/87.6	0	34.7	$8.2\cdot10^{-3}$	32.1	$4\cdot10^{-3}$
Case 2							
0	–	116.7/219.5	$7.3\cdot10^{-2}$	32.4	$1.7\cdot10^{-3}$	92.9	$9.1\cdot10^{-2}$
18	0.8/0.2	110.1/217	$6\cdot10^{-2}$	32.6	$1.9\cdot10^{-3}$	92.5	$9.2\cdot10^{-2}$
36	0.88/0.91	51.9/136.1	$7.7\cdot10^{-3}$	34.2	$4.2\cdot10^{-3}$	89.6	$9.4\cdot10^{-2}$
54	0.92/1	18.2/123.6	$3.6\cdot10^{-5}$	34.7	$8.2\cdot10^{-3}$	79.2	10^{-1}
Case 3							
0	–/–	114.5/216.6	$6.7\cdot10^{-2}$	92.9	$9.3\cdot10^{-2}$	30.1	$1.2\cdot10^{-3}$
18	$0.95/10^{-3}$	107.1/216.6	$1.9\cdot10^{-2}$	91.7	$9.4\cdot10^{-2}$	30.5	$1.3\cdot10^{-3}$
36	1/0.51	25.8/139.3	$1.9\cdot10^{-4}$	80.1	$1\cdot10^{-1}$	31.8	$1.7\cdot10^{-3}$
54	1/0.71	16.1/99.5	0	70.5	$1.2\cdot10^{-1}$	32.3	$3.2\cdot10^{-2}$

p_s—the probability that D-packet replaces R-packet in the queue, p_{loss}—probability that packet is lost, D_m—mean packet transfer delay, $D_{\max}$—maximum packet transfer delay.

Table 2
End-to-end B- and D-packet transfer quality versus R-packet rate (V_R)

Generation rate	Node N1/N2	PFS flow S1-D1	Non-PFS flow S2-D2	Non-PFS flow S3-D3
V_R [kbit/s]	p_s	T [s]/G [kbit/s]	D_m [ms]/p_{loss}	D_m [ms]/p_{loss}
0	–/–	230.0/347.8	$56.0/8.3\cdot10^{-3}$	$54.3/6.1\cdot10^{-3}$
13	0.52/0.06	213.9/374.0	$62.5/1.6\cdot10^{-2}$	$60.3/1.1\cdot10^{-2}$
26	0.91/0.14	185.2/432.0	$80.0/3.6\cdot10^{-2}$	$82.0/2.8\cdot10^{-2}$
52	0.98/0.52	170.8/468.4	$75.3/9.2\cdot10^{-2}$	$84.2/7.1\cdot10^{-2}$

p_s—the probability that D-packet replaces R-packet in the queue, p_{loss}—probability that packet is lost, D_m—mean packet transfer delay, T—file upload time, G—TCP goodput.

than 150 ms, for packet loss ratio—less than 10^{-4}. Table 1 shows the received results, corresponding to the Cases 1, 2 and 3, illustrating the quality experienced by VoIP packets supported by PFS and without PFS, versus generation rate (constant) of R-packets (V_R). Packet size for R-packets was fixed to 28 bytes. Notice that by adding R-packets, we increase the total system load.

The presented results show that quality of VoIP application in the cases, when the IP network is under heavy load conditions ($\rho = 0.95$) is non acceptable. The packet loss rate is greater than 10^{-3} while maximum packet delay is greater than 200 ms. Anyway, by using PFS we may get acceptable quality, even if the network is overloaded ($\rho = 1.1$). Obviously, this requires greater R-packet emitting rate, which is almost 36 kbit/s (Table 1).

5.2. FTP application

In this section we show the usefulness of using PFS mechanism for getting better quality in the case of FTP application. The FTP is sending traffic using TCP protocol. This traffic is submitted, as in Section 5.1., to the tested network with 3 routers, as depicted in Fig. 6. The rates of inter-router and access links as well as the buffers of output router ports are the same as in Section 5.1.

The tested FTP connection supported by PFS mechanism is established between S1-D1 and passes the routers N1, N2 and N3. FTP client uses this connection to upload 10 Mbit file on FTP server. S2-D2 and S3-D3 constitute background traffic, each generated according to Poissonian law with the mean rate 1.5 Mbit/s and constant packet size 750 bytes, for

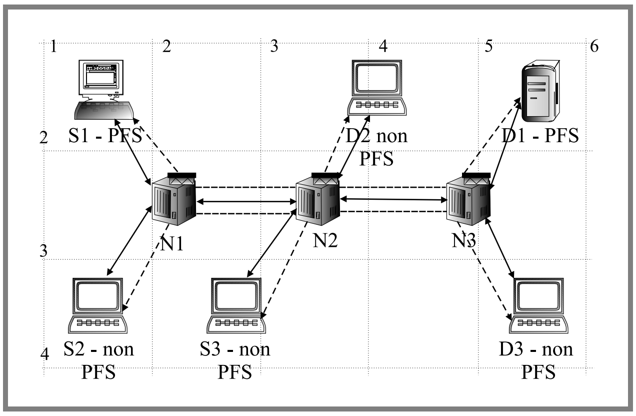

Fig. 6. Network topology for testing FTP application.

getting independent load conditions on the links N1↔N2 and N2↔N3.

Table 2 shows the received values of TCP upload file time/goodput characteristics in the case of FTP user and packet transfer characteristics (mean packet delay and packet loss rate) for Poissonian back-ground traffic, as a function of R-packet rate (V_R). R-packets are of 40 bytes each.

The presented results show, that by using PFS mechanism we can improve file upload time for FTP. Again, by increasing R-packets rate the TCP goodput is also increasing.

6. Conclusions

In the paper we presented the recent obtained results corresponding to efficiency of the PFS mechanism. Comparing to the [6], simple analysis of system with PFS was introduced and the results illustrating possibility of shaping packet delay characteristics for PFS flows were shown. Furthermore, we examined VoIP and FTP application, using PFS for improving end-to-end quality. It appeared that in both considered cases we can obtain satisfactory quality, even if the network is overloaded. Further studies are focused on detailed system analysis and implementations issues.

References

[1] A. Bąk, W. Burakowski, F. Ricciato, S. Salsano, and H. Tarasiuk, "Traffic handling in AQUILA QoS IP network", in *Proc. Quality of Future Internet Services, QoFIS 2001, Lecture Notes in Computer Science*. Springer, 2001, vol. 2156.

[2] A. Bąk *et al.*, "AQUILA network architecture: first trial experiments", *J. Telecommun. Inform. Technol.*, no. 2, 2002, p. 3–13.

[3] Y. Bernet *et al.*, "Differentiated services", Internet Draft, Feb. 1999, draft-ietf-diffserv-framework-0.2.txt

[4] S. Blake *et al.*, "An architecture for differentiated services", Internet RFC 2475, Dec. 1998.

[5] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview", RFC 1633, June 1994.

[6] W. Burakowski and M. Fudała, "Priority forcing scheme: a new strategy for getting better than best effort service in IP-based network", in *Internet Technol., Appl. Soc. Imp.* Kluwer, 2002.

[7] *Broadband Network Teletraffic. Performance Evaluation and Design of Broadband Multiservice Networks. Final Report of Action COST 242*, J. Roberts, U. Mocci, and J. Virtamo, Eds., *Lecture Notes in Computer Science*. Heidelberg: Springer, 1996, vol. 1155.

[8] D. Gross and C. Harris, *Fundamentals of Queuing Theory*, 3 rd. ed., Series in Probability and Statistics. Boston: Wiley, 1998.

[9] P. Hurley, J. Y. Le Boudec, P. Thiran, and M. Kara, "ABE: providing a low-delay service within best effort", *IEEE Network*, no. 3, pp. 60–69, 2001.

[10] S. Keshav, *An Engineering Approach to Computer Networking.* Addison-Wesley, 1997, Chap. 9.

[11] D. Wu *et al.*, "Streaming video over the Internet: approaches and directions", *IEEE Trans. Circ. Syst. Video Technol.*, March 2001.

Monika Fudała was born in Poland in 1976. She received her M.Sc. degree in telecommunications from Warsaw University of Technology in 2000. She is now a Ph.D. student. Her research interest focus mainly on traffic handling mechanisms in QoS IP networks.

e-mail: mkrol@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Wojciech Burakowski was born in Poland in 1951. He received his M.Sc., Ph.D. and D.Sc. degrees in telecommunications from Warsaw University of Technology, in 1975, 1982 and 1992, respectively. He is now a Professor in Warsaw University of Technology, Institute of Telecommunications. Since 1990, he has been involved in the European projects COST 224, COST 242, COST 257, COST 279 and AQUILA. His research interests include performance evaluation of ATM and IP networks.

e-mail: wojtek@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland