The Impact of Congestion Control Mechanisms on Network Performance after Failure in Flow-Aware Networks

Jerzy Domżał, *Student Member, IEEE*, Robert Wójcik and Andrzej Jajszczyk, *Fellow, IEEE*

***Abstract*—This paper presents the impact of congestion control mechanisms proposed for Flow-Aware Networks on packet transmission in the overloaded network after a link failure. The results of simulation based analysis show how to set the values of the congestion control parameters in order to decrease the acceptance time of the interrupted streaming flows in the backup link. The research was performed for three congestion control mechanisms, the Enhanced Flushing Mechanism (EFM), the Remove Active Elastic Flows (RAEF), and Remove and Block Active Elastic Flows (RBAEF) in two different cross-protect router architectures, with the PFQ (Priority Fair Queuing) and with the PDRR (Priority Deficit Round Robin) scheduling algorithms. Moreover, the advantages and weaknesses of using the proposed solutions in FAN, considering the effects of a network element failure, are described and analyzed.**

***Index Terms*—Congestion Control, Flow-Aware Networks, Protection, Restoration.**

I. Introduction

The Quality of Service for packet networks is still a very important and interesting issue. The dominating QoS architecture in the current networks is DiffServ (Differentiated Services). Unfortunately, this architecture is complicated and in many cases does not work as expected. Currently, high quality data transmission is possible only because of overprovisioned network links. The more and more popular applications like VoIP or VoD need low delays, low packet loss and link capacities high enough to work satisfactorily. In fact, all new applications and services in packet networks require a controlled quality of connections. This necessity has triggered many studies on providing new possibilities of ensuring the proper Quality of Service in the packet networks. Flow-Aware Networking (FAN) is a new concept for packet switched networks with QoS guaranties. The main assumption of FAN is to provide maximum possible benefits in the perceived QoS using only the minimal knowledge of the network. In this paper, we argue that by using proper congestion control mechanisms in FAN, we can enhance the perceived QoS, especially in case of network failures.

The remainder of the document is organized as follows. Section 2 introduces the general idea of FAN. Section 3 shows the congestion control mechanisms for FAN and their brief description. In Section 4, the results of carefully selected simulation experiments for each congestion control algorithm and two different queuing disciplines are presented. Section 5 concludes the paper.

J. Domżał, R. Wójcik and A. Jajszczyk are with the Department of Telecommunications, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland (e-mail: {domzal, wojcik, jajszczyk}@kt.agh.edu.pl).

II. Flow-Aware Networks

The concept of Flow-Aware Networking as a novel approach to assure quality of service in packet networks was introduced in 2004 [1]. The goal of FAN is to enhance the current IP network by improving its performance under heavy congestion. To achieve that, certain traffic management mechanisms to control link sharing are introduced, namely: measurement-based admission control [2] and priority scheduling [1], [3]. The former is used to keep the flow rates sufficiently high, to provide a minimal level of performance for each flow in case of overload. The latter realizes fair sharing of link bandwidth, while ensuring negligible packet latency for flows emitting at lower rates.

In FAN, admission control and service differentiation are implicit. There is no need for a priori traffic specification, as well as there is no class of service distinction. However, streaming and elastic flows are implicitly identified inside the FAN network. This classification is based solely on the current flow peak rate. All flows emitting at lower rates than the current fair rate are referred to as streaming flows, and packets of those flows are prioritized. The remaining flows are referred to as elastic flows. The distinctive advantage of FAN is that both streaming and elastic flows achieve a necessary quality of service without any mutual detrimental effect.

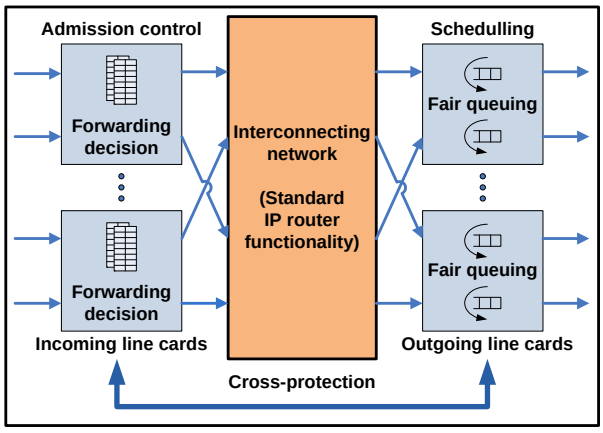

Fig. 1. Concept diagram of a Cross-Protect router [1]

A. The Cross-Protect Mechanism

FAN is supposed to be an enhancement of the existing IP network. In order to function properly, an upgrade of current IP routers is required. Figure 1 shows a concept diagram of a Cross-Protect router (XP router), the interconnecting device in FAN networks. FAN adds only two blocks to the standard IP router. They are namely: admission control block and scheduling block. The former is placed in the incoming line cards of the router, whereas the latter is situated in the outgoing line cards.

Admission control is responsible for accepting or rejecting the incoming packets, based on the current congestion status. If a packet is allowed, the identifier (ID) of flow associated with it may be added to the protected flow list (PFL), and then all forthcoming packets of this flow will be accepted. The packets of new flows may be accepted in the admission control block only when the links are not congested. The ID is removed from the PFL after a specified time period of flow inactivity given by the value of the *pfl_flow_timeout* parameter. The admission control block realizes the measurement based admission control (MBAC) functionality [4]. The scheduler is responsible for queue management and it has to ascertain that all flows are equally treated.

Naming FAN devices as "Cross-Protect routers" is a result of a mutual cooperation and protection, which exists between both discussed blocks. The admission control block limits the number of active flows in the XP router, which essentially improves the queuing algorithm functionality, and reduces its performance requirements. It is vital that queuing mechanisms operate quickly, as for extremely high speed links the available processing time is strictly limited. On the other hand, the scheduling block provides admission control with the information on congestion status on the outgoing interfaces. The information is derived based on, for example, current queues occupancy. The cross-protection contributes to a shorter required flow list and queue sizes, which significantly improves FAN scalability.

B. Scheduling

A queuing algorithm, implemented in the scheduler block is the most important mechanism when considering the congestion control issue. It decides (by measurements) which flows may be served and which should be blocked. It allows for fair access to the resources without any intervention form the user. Over time, numerous queuing algorithms have been proposed [5]. In our analysis, we used FAN links with the implemented functionality of the PFQ (Priority Fair Queuing) and PDRR (Priority Deficit Round Robin) fair queuing algorithms.

PFQ is a modified version of the SFQ (Start-time Fair Queuing) algorithm [6]. PFQ inherits the advantages of SFQ and is enriched by the packet prioritizing possibilities in the scheduler module. Similarly, PDRR is an enhanced version of DRR (Deficit Round Robin) [7]. These algorithms operate differently, however, the outcome of their functioning is almost identical. A more detailed description of both queuing disciplines, including pseudocodes, measured indicators and all required definitions may be found in [1] and [3].

These algorithms implicitly give priority to the packets of flows whose peak rate is less than the current fair rate. The flows with rates less than the current fair rate are assigned high priority. This way, streaming flows with peak rates less than the current fair rate are subjects to the bufferless multiplexing and, therefore, perceive low delays and losses [8].

To provide the admission control block with proper congestion status, *priority_load* and *fair_rate* indicators are measured periodically by the scheduling block. The *priority_load* represents the sum of the lengths of priority packets transmitted in a certain time interval, divided by the duration of that interval, and normalized with respect to the link capacity. The *fair_rate* indicates approximately the throughput achieved by any flow that is continuously backlogged. In other words, it is the rate available to each flow at the moment. The detailed description of both *fair_rate* and *priority_load* parameters, along with the methods of estimating them are presented in [1] and [3].

Both congestion indicators are calculated periodically. As mentioned before, the *fair_rate* is used to differentiate between streaming and elastic flows within the XP router. Additionally, along with the *priority_load*, it is used by the admission control to selectively block new incoming flows, provided that the congestion state is detected.

III. Congestion Control Mechanisms for FAN

The congestion control mechanisms for FAN, presented in [9] and [10], were proposed to decrease the access time to the congested link for new streaming flows that may represent, for example, the VoIP connections. Users expect that after having dialed the number, their connection will go through almost immediately. Therefore, it is necessary to provide short acceptance times for such flows. The acceptable time for international calls should be not greater that 11 seconds for the 95% of the calls, while for the local calls it should not exceed 6 seconds [11]. In the basic version of the admission control algorithm, the new flows cannot be accepted in the XP routers under the state of congestion. It may cause that, in some cases, the whole bandwidth may be occupied by a finite number of flows giving no chance for new flows to begin their transmission for a long time. The new mechanisms based on the whole or partial cleaning the PFL content in the congestion state allow for decreasing the acceptance time of new streaming flows. It is possible to choose such values of the parameters characteristic for the proposed solutions that result in achieving a short acceptance time of new streaming flows, but not significantly increasing the value of the mean transmission time of elastic flows. Three versions of congestion control mechanisms; EFM, RAEF, and RBAEF are proposed and described below in details. The main goal of this paper, however, is to show that these mechanisms also work very well when a network device or a link fails and the traffic redirection is needed.

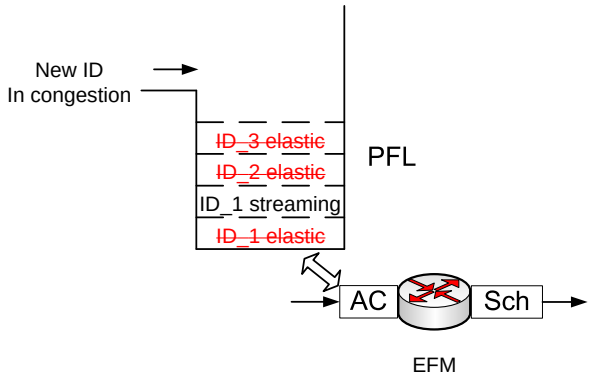

Fig. 2. The operation principle of EFM

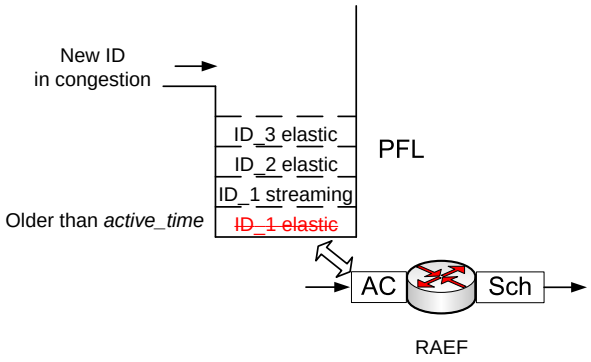

Fig. 3. The operation principle of RAEF

A. The Enhanced Flushing Mechanism

The EFM (Enhanced Flushing Mechanism), presented in [9] and [10], is a good proposition to solve the problem of too long acceptance times of new streaming flows in the AC block. In this algorithm the identifiers of all elastic flows are removed from the PFL if a packet of a new flow comes to the AC block in the congestion state (see Figure 2). However, the PFL should not be flushed (erased) in all such cases. It is necessary to provide a time variable denoted by us as *pfl_flushing_timer*. The value of this variable represents the minimum period of time that has to expire before the next flushing action can occur. It ensures a stable operation of the algorithm. The flows which identifiers were removed from the PFL are not blocked in the AC block and can resume the transmission promptly. After removing the identifiers of the elastic flows from the PFL, the flows have to compete with each other for acceptance in the AC block and it may take some time before they will be accepted again.

Based on the results obtained in the experiment described in [10] we can conclude that it is possible to ensure short acceptance times of new streaming flows in the AC block independently of the number of elastic flows being active in the background and the number of streaming flows, which want to begin the transmission. The EFM works sufficiently well with both the PFQ and the PDRR algorithms.

The implementation of the EFM in the cross-protect router is quite simple and does not increase the complexity and power resources significantly.

B. The RAEF Mechanism

The second mechanism proposed to solve the problem of the too long acceptance time of new streaming flows in the AC block is RAEF (Remove Active Elastic Flows). In this algorithm, only the identifiers of those elastic flows that were active for at least a specified period of time (*active_time*) are removed from the PFL when congestion is noticed (see Figure 3). The flows which identifiers were removed from the PFL are not blocked in the AC block (as in the EFM) and can resume the transmission promptly. The disadvantages of this algorithm are also the same as in the EFM. It is possible that the identifiers of such flows will not be added to the PFL again immediately or even in a short time and the transmission time of their traffic may be extended. The flows which identifiers were removed from the PFL, have to compete with other flows for acceptance in the AC block and it may take some time before such flows will be accepted again.

The results of carefully selected simulation experiments for analyzing the RAEF mechanism are presented in [9]. They show that the algorithm ensures quick acceptance times of new streaming flows in the AC block independently of the number of elastic flows being active in the background and the number of streaming flows which want to begin the transmission. Similarly to EFM, the RAEF mechanism works satisfactorily with both analyzed versions of the scheduling algorithm, the PFQ and the PDRR.

Similarly as in case of EFM, the implementation of the RAEF mechanism in the cross-protect router is simple and does not increase the complexity and power resources significantly.

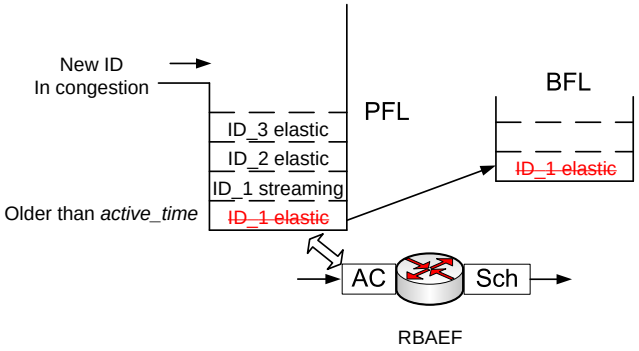

Fig. 4. The operation principle of RBAEF

C. The RBAEF Mechanism

The last mechanism proposed in [9] to decrease the time interval between beginning of sending the packets by a new streaming flow and its acceptance in the AC block is called RBAEF (Remove and Block Active Elastic Flows). In this algorithm the identifiers of the elastic flows being active for a specified period of time are removed from the PFL

every time when congestion is noticed — just as in case of RAEF. However, the identifiers of such flows are then written to the BFL (Blocked Flow List) for a short, fixed period of time called *blocked_time* (see Figure 4). If a packet arriving to the admission control block belongs to the flow, the identifier of which is in the BFL, the packet is always dropped. Therefore, the flows removed from the PFL list can continue the transmission only after their tag has been removed from the BFL. The flows which identifiers were removed from the BFL, can continue transmission, but again, they have to compete with other flows for link resources and it may take some time before such flows will be accepted again.

The simulation scenario for analyzing the RBAEF mechanism is described in [9]. Based on the results obtained in the experiment we can conclude that this algorithm also ensures quick acceptance of new streaming flows in the AC block independently of the number of elastic flows being active in the background and the number of streaming flows which want to begin the transmission. As both previously presented mechanisms, the RBAEF mechanism also works satisfactorily with both analyzed versions of the scheduling algorithm, the PFQ and the PDRR.

The implementation of the RBAEF mechanism in the cross-protect router is slightly more complicated than in the previous cases, but does not increase the complexity and power resources significantly, either.

IV. FAN in Case of Failure

The analysis of traffic in the network is very important, especially in the case of failures. In the basic FAN architecture, the ID of a flow can be removed from the PFL list only in case of its long enough inactivity. It means that the transmission of all flows accepted in the AC block cannot be stopped and it is true until there are no failures in the network.

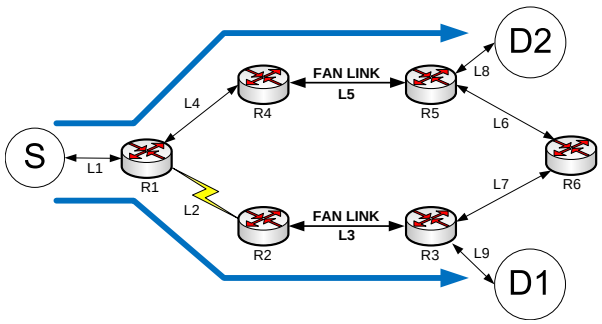

Fig. 5. Simulated network topology

In this section we analyze the mean acceptance time of a new streaming flow in the topology presented in Figure 5. There is one source node and two destination nodes in our experiments. We assumed that bottleneck links L3 and L5 are FAN links of the 100 Mbit/s capacity. The capacity of the rest of the links, with the FIFO queue, was set to 1 Gbit/s. The shortest path routing was implemented in this network, which means that under normal conditions the traffic to node D1 is sent through nodes R1, R2 and R3 while the traffic to node D2 is sent through nodes R1, R4 and R5. By using such a topology we decided that link L5 is treated as a backup for the traffic sent normally through the link L3. We analyzed the effects of failures of link L2 at a chosen time instant.

We provided the traffic pattern with the Pareto distribution for calculating the volume of the elastic traffic directed to both destination nodes. The exponential distribution for generating the time intervals between beginnings of the transmissions of the elastic flows was used. The exponential distribution was also used to generate the start times of 20 streaming flows. The packet size (100 bytes) and the transmission rate (80 kbit/s) used for the streaming flows are the typical values of the VoIP stream transmission, e.g., in Skype. We made our simulation runs in various conditions changing the number of elastic and streaming flows. We analyzed the acceptance time of each streaming flow in the AC block of node R2 (before failure) and node R4 (after failure). The measurement interval for the *priority_load* parameter was set to 50 ms while the *fair_rate* values were estimated every 500 ms. The *max_priority_load* (maximum allowed value of the priority load) and the *min_fair_rate* (minimum allowed value of the fair rate) parameters were set to 70% and 5% of the link capacity, respectively, and the *pfl_flow_timeout* parameter was set to 20 s. 95% confidence intervals were calculated by using the Student's t-distribution.

The simulations were performed under various conditions. Firstly, we decided to check the mean acceptance time of new streaming flows in FAN links with the number of background elastic flows ranging from 200 to 600. The duration of each simulation run was set to 500 s. At 250 s, link L2 was turned off and the packets of all flows were sent through link L5. The redirected streaming flows had to compete for access to the L5 link along with all other elastic flows.

Basic FAN links have unacceptable values of the *waiting_time* (see Table I). It takes tens of seconds (and even hundreds of seconds for the redirected flows) before a new flow is accepted in the router. If we imagine that this flow exemplifies a VoIP call, it is obvious that the break in transmission of such a flow is much too long. Our simulations show that the acceptance time of streaming flows can be decreased by using the congestion control mechanisms presented in [9] on both examined FAN links. The comparison of the *waiting_time* values for the basic FAN with PFQ algorithm and for its modified versions with EFM (*pfl_flushing_timer* = 5 s), RAEF (*active_time* = 5 s) and RBAEF (*active_time* = 5 s, *blocked_time* = 1 s) are presented in Figure 6. We can see that difference between the values for basic FAN and its modified versions are significant. If we use the mentioned above congestion control mechanisms the new streaming flows may be accepted in the routers after less then one second in both the basic and backup links.

The presented results show that the acceptance time of new streaming flows do not depend on the number of elastic flows in the background. The simulation results for the case with 200 elastic flows are presented in Table I. In basic FAN, for both queuing algorithms, a new streaming flow is accepted in the

TABLE I
TIME INTERVAL BETWEEN BEGINNING OF SENDING THE PACKETS AND THE ACCEPTANCE OF A STREAMING FLOW IN THE AC BLOCK IN AN XP ROUTER BEFORE AND AFTER L2 LINK FAILURE

Mechanism	PFQ		PDRR	
	waiting time router R2 [s]	waiting time router R4 [s]	waiting time router R2 [s]	waiting time router R4 [s]
Basic FAN	78.43 ± 6.56	240.86 ± 26.64	89.28 ± 7.87	258.34 ± 30.65
EFM	0.46 ± 0.10	0.67 ± 0.22	0.98 ± 0.38	0.54 ± 0.11
RAEF	$0.37 \pm .023$	0.45 ± 0.13	0.15 ± 0.10	0.82 ± 0.67
RBAEF	0.56 ± 0.21	0.90 ± 0.23	0.05 ± 0.01	0.08 ± 0.06

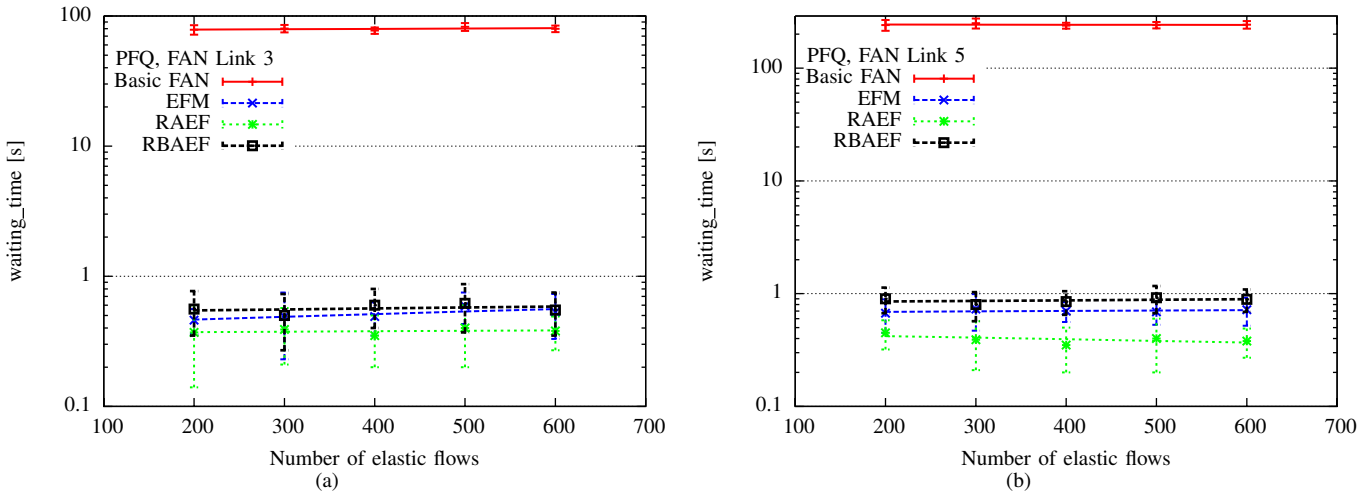

Fig. 6. The mean acceptance time of new streaming flows in basic FAN and its versions with EFM, RAEF and RBAEF, (a) for Link L3, (b) for Link L5

AC block of node R2 after tens of seconds while after L2 link failure this time (observed at node R4) raises to a few hundreds seconds. The *waiting_time* period in the R4 router represents the amount of time in which the streaming flow struggles for acquiring the backup link's resources. The difference between acceptance time of a new streaming flow in routers R2 and R4 is significant. A new flow may be accepted in router R2 when one or more elastic flows finish their transmission which allows for increasing the *fair_rate* values. After a failure, all redirected flows have to compete for the access to the R4 router which significantly increases the number of competitors in that node. Successively, this decreases their chance for being accepted and, therefore, increases the mean acceptance time.

The mean *waiting_time* values for all three congestion control mechanism and both queuing disciplines are presented in Table I and in Figures 7, 8 and 9. The values of *pfl_flushing_timer* and *active_time* parameters were set to 5 s, while the *blocked_time* threshold in RBAEF was set to 1 s.

In the EFM mechanism (Figure 7) we present the mean waiting time with respect to the *pfl_flushing_timer* parameter. This parameter defines the minimum time period that has to expire before a new flushing can occur. Under normal circumstances, a new streaming flow has to wait for the congestion to end, however, flushing mechanism allows for faster acceptance of such flows. By using EFM, a flow must only wait for the nearest flushing procedure. Therefore, a statistic streaming flow will only need a half of the *pfl_flushing_timer* to be accepted on a link. This explains the linear growth of the dependency presented in Figure 7.

Figures 8 and 9 present similar dependencies but in case of these flushing mechanisms, the *pfl_flushing_timer* is replaced by the *active_time* parameter. Now, the PFL may be flushed each time a new flow requires it, however, only the flows that were active longer than the *active_time* are erased. By analogy, we can observe that, statistically, a new streaming flow must wait for no longer than a half of the *active_time*. Moreover, the mentioned dependencies are similar for all flushing mechanisms and for both queuing disciplines.

Based on the obtained results, we may conclude that the congestion control mechanisms proposed for FAN significantly improve the performance of traffic classified as streaming. It is possible to set the values of the *active_time* (for RAEF and RBAEF) and the *pfl_flushing_timer* (for EFM) parameters so that they ensure very short acceptance time of streaming flows in a backup link after a failure. It is a strong advantage of these congestion control mechanisms. There are, however, some drawbacks of the proposed solution. The most important one is that the transmission of some elastic flows may be ceased by each flushing procedure. Therefore, it may increase the total transmission time of those flows. However, in [9], it is shown that the mean total transmission time of the elastic flows does not necessarily increase by using these congestion control mechanisms. It is possible to set the transmission parameters to such values, that allow for a significant decrease

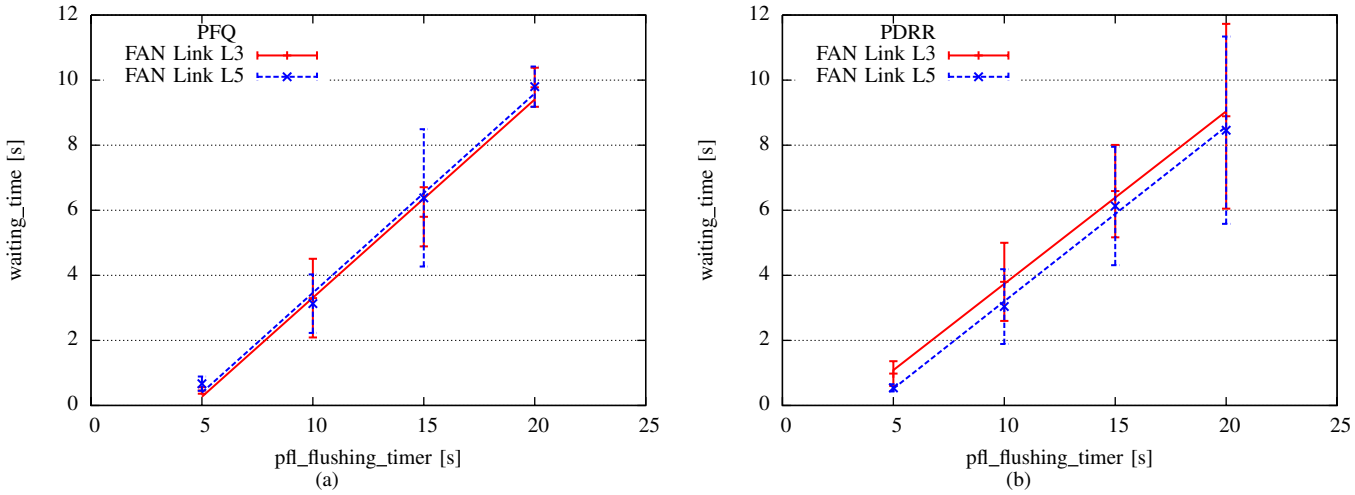

Fig. 7. The mean acceptance time of new streaming flows in the FAN nodes with EFM, (a) with PFQ scheduling algorithm, (b) with PDRR scheduling algorithm

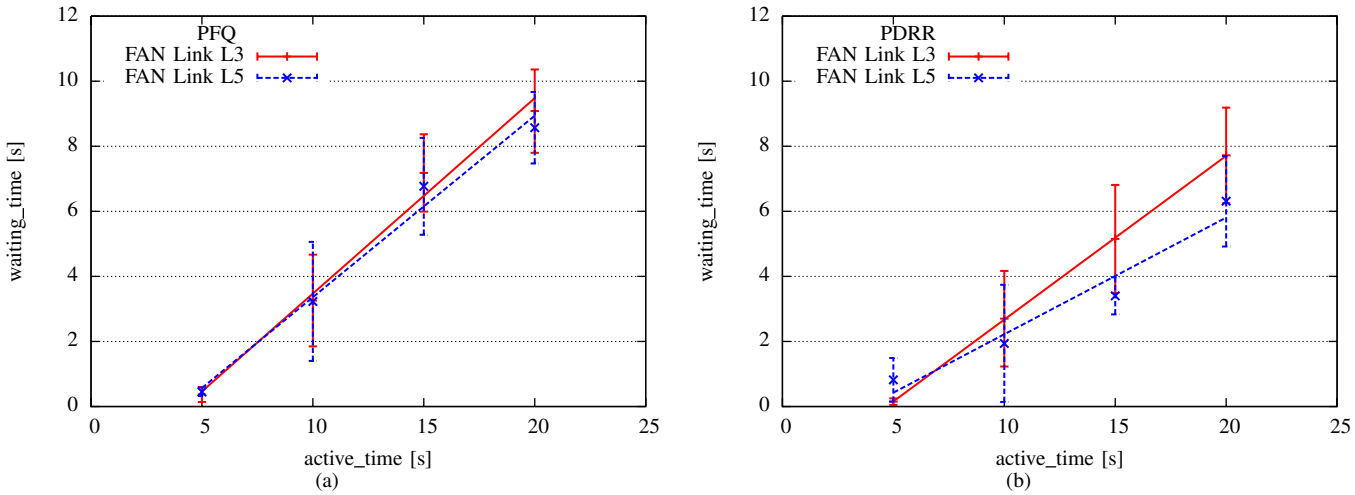

Fig. 8. The mean acceptance time of new streaming flows in the FAN nodes with RAEF, (a) with PFQ scheduling algorithm, (b) with PDRR scheduling algorithm

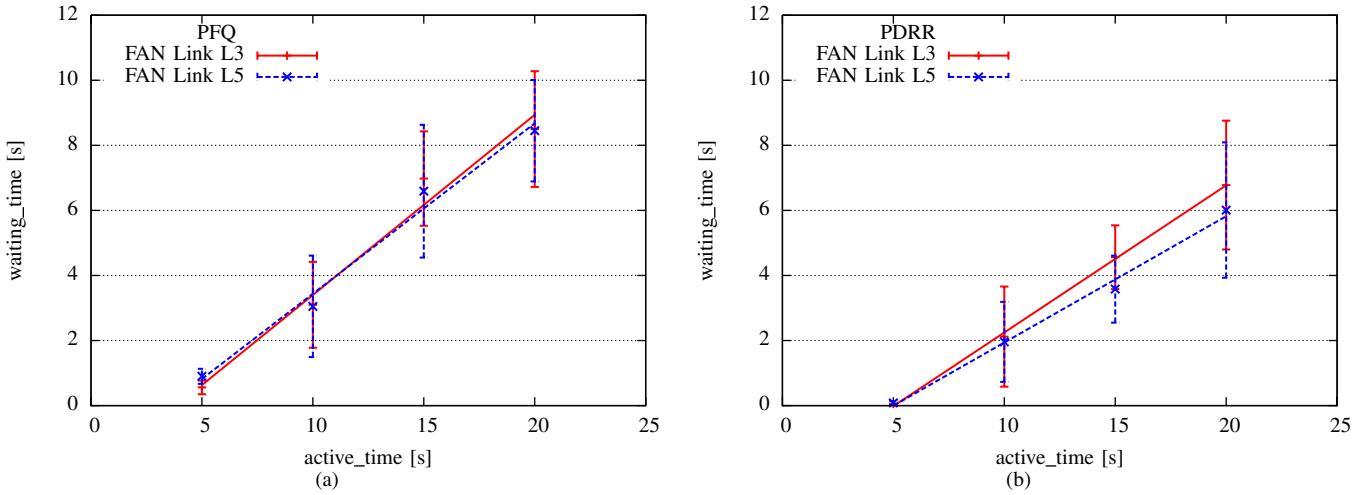

Fig. 9. The mean acceptance time of new streaming flows in the FAN nodes with RBAEF, (a) with PFQ scheduling algorithm, (b) with PDRR scheduling algorithm

of the acceptance times of all the streaming flows without increasing the mean total transmission time of the elastic flows significantly.

V. Conclusion

The traffic management in FAN is described in this paper. In the congestion-less state, a new flow is always accepted in the AC block of an XP router while, under congestion, no new IDs can be added to the PFL. Three congestion control mechanisms, namely: EFM, RAEF and RBAEF that allow for fast acceptance of new streaming flows in the FAN router are briefly introduced. By analyzing the traffic in a simple FAN topology it is shown that these congestion control mechanisms significantly decrease the waiting time for all streaming flows. Moreover, they allow for fast acceptance of the streaming flows in the backup link after primary link's failure, which enhances FAN's network restoration capabilities. It is important to note that the backup link may also be used for normal transmission in the failure-less state and, after a primary link's failure, streaming flows from that link can be transferred via the backup link with hardly any interruptions. The IDs of the elastic flows in the backup link are removed from the PFL and have to compete for access to the resources with all flows that want to send the packets, including the flows from the broken primary link.

The solutions proposed in the paper are flexible, stable and give the chance for more reliable transmission in Flow-Aware Networks.

Acknowledgement

This work was done within the EU FP7 NoE Euro-NF (http://www.euronf.org) framework. The reported work was also supported by the Foundation for Polish Science and the Polish Ministry of Science and Higher Education under grant N517 013 32/2131.

References

[1] A. Kortebi, S. Oueslati, and J. Roberts, "Cross-protect: implicit service differentiation and admission control," in *IEEE HPSR 2004*, Phoenix, USA, April 2004.

[2] S. Oueslati and J. Roberts, "A new direction for quality of service: Flow-aware networking," in *NGI 2005*, Rome, Italy, April 2005.

[3] A. Kortebi, S. Oueslati, and J. Roberts, "Implicit Service Differentiation using Deficit Round Robin," in *ITC19*, Beijing, China, August/September 2005.

[4] ——, "MBAC algorithms for streaming flows in Cross-protect," in *EuroNGI Workshop*, Lund, Sweden, June 2004.

[5] B. Suter, T. V. Lakshman, D. Stiliadis, and A. K. Choudhury, "Buffer management schemes for supporting TCP in gigabit routers with per flow queuing," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1159–1169, June 1999.

[6] P. Goyal, H. M. Vin, and H. Cheng, "Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 690–704, October 1997.

[7] M. Shreedhar and G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 375–385, June 1996.

[8] J. Roberts, "Internet Traffic, QoS and Pricing," *Proceedings of the IEEE*, vol. 92, pp. 1389–1399, September 2004.

[9] J. Domzal and A. Jajszczyk, "New Congestion Control Mechanisms for Flow-Aware Networks," in *IEEE ICC*, Beijing, China, May 2008.

[10] ——, "The Flushing Mechanism for MBAC in Flow-Aware Networks," in *NGI 2008*, Krakow, Poland, April 2008.

[11] ITU-T E.721 Network grade of service parameters and target values for circuit-switched services in the evolving ISDN.