Reliable Transmission in Flow-Aware Networks

Jerzy Domżał[1], Robert Wójcik[1], Andrzej Jajszczyk[2]
[1]*Student Member, IEEE*, [2]*Fellow, IEEE*

***Abstract*—A complete system which ensures reliable transmission of streaming flows in Flow-Aware Networks (FAN) is presented and analyzed in the paper. A new congestion control mechanism, called RPAEF (Remove and Prioritize in access Active Elastic Flows), is described in details. It allows for fast acceptance of new streaming flows in the admission control (AC) block of FAN routers. The mechanism of limiting the number of new flows accepted in the AC block is provided as an answer to the scalability problem of all congestion control mechanisms proposed for FAN up to now. The global list of protected flows ensures that all streaming flows redirected from the primary route in case of a network element failure are immediately accepted in the first router on a backup route. The advantages and weaknesses of the proposed solutions are described and analyzed. Moreover, it is shown that the simultaneous implementation of all of them ensures fast, scalable and reliable transmission of streaming flows in FAN.**

***Index Terms*—Flow-Aware Networks; traffic engineering; Quality of Service; congestion control; reliability**

I. Introduction

The Flow-Aware Networking concept was introduced by James Roberts and Sara Oueslati from France Telecom in [1] and, then, presented as a complete system in 2004 [2]. The main goal of this proposal is to ensure the proper quality of service (QoS) in packet networks in an implicit way. In comparison with the well known QoS architectures, like DiffServ (*Differentiated Services*) [3] or IntServ (*Integrated Services*) [4], FAN is easier to implement and conforms to net neutrality paradigms. The main assumption of FAN is the provision of maximum possible benefits in the perceived QoS by using only the minimal knowledge of the network. However, there are still some problems which have to be solved when considering a reliable transmission in FAN. In this paper, we propose a complete system which ensures good scalability as well as fast acceptance times and uninterrupted transmission of streaming flows even after a network element failure.

This paper is organized as follows. Section II describes the FAN concept. Section III shows the assumptions and the description of a new congestion control mechanism, called RPAEF. In Section IV, the mechanism for limiting the number of new flows accepted in the FAN routers is presented. The concept of the GPFL (*Global Protected Flow List*) is analyzed in Section V. The results of carefully selected simulation experiments for each new solution are presented at the ends of the related sections. Section VI 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: {jdomzal, wojcik, jajszczyk}@kt.agh.edu.pl).

II. Flow-Aware Networks

The main assumption of the Flow-Aware Networking concept is to achieve efficient packet transmission in a simple way and with the minimal knowledge of the network. All the traffic in FAN is sent as flows and served in the cross-protect routers (also denoted as XP's) which are the basic elements of the FAN architecture. There are two traffic types defined in FAN:

- *elastic* — usually used for data transmission, served with the best effort regime;
- *streaming* — used for low bandwidth consuming services, e.g., VoIP calls, served with priority over the elastic type.

All the flows are implicitly classified to one of the presented types based on their bandwidth occupation and, then, served with (streaming) or without (elastic) priority.

There are two main functional blocks in the XP routers. The admission control (AC) decides of accepting or rejecting the packets of flows while the scheduler is responsible for periodical measuring the values of the following two parameters:

- *fair_rate* — estimates the maximum rate that might be or is realized by elastic flows;
- *priority_load* — measured as a quotient of the sum of the queued packet lengths (with priority) in a given time period to the length of this period.

In the congestion-less state, new flows are accepted in the AC block and their identifiers (IDs) are written to the PFL (*Protected Flow List*). The ID of a flow is removed from the PFL if the flow is inactive for a fixed time period given by the value of the *pfl_flow_timeout* parameter. Each outgoing link connected to the FAN router has its own PFL.

A FAN link is congested if the value of *fair_rate* is lower than the *min_fair_rate* (minimum allowed value of the *fair_rate*) or the value of *priority_load* is higher than the *max_priority_load* (maximum allowed value of the *priority_load*). We have to note that in almost every case the congestion is indicated by the *fair_rate* parameter. In congestion, only the packets of flows which IDs are written to the PFL may be served. It means that sometimes a new flow has to wait for a long time before it is allowed to begin transmission. This situation is unacceptable for the real-time applications, like e.g., VoIP (*Voice over IP*) or VoD (*Video on Demand*).

Two scheduling algorithms were proposed to be used in FAN, PFQ (*Priority Fair Queuing*) or PDRR (*Priority Deficit Round Robin*). Because of lack of space we omit the description of them. It can, however, be found in [2] and [5]. As the solutions proposed in this paper do not modify the scheduling block, the simulation analysis presented in this

978-1-4244-4148-8/09/$25.00 ©2009

paper is provided only for the PFQ algorithm. The results of the same analysis for FAN with the PDRR are similar to those obtained for FAN with the PFQ.

FAN is scalable since the complexity of queuing algorithms does not increase with the link capacity [6]. Moreover, fair queuing is feasible, as long as link load is not allowed to attain saturation levels, which is asserted by the admission control. Compared to other QoS architectures, FAN scalability, due to the lack of signaling and very low data handling complexity, is not matched by any other architecture [7]. Finally, FAN is a solution which conforms to net neutrality paradigms, as the differentiation is based only on the internal, implicit node decisions. This way, services in a network may be differentiated, while the fairness and neutrality is maintained.

III. The RPAEF mechanism

There are three congestion control mechanisms for FAN described and analyzed in [8], [9] and [10]. The EFM (*Enhanced Flushing Mechanism*), RAEF (*Remove Active Elastic Flows*) and RBAEF (*Remove and Block Active Elastic Flows*) work based on total or partial cleaning of the PFL content in the AC block in congestion. It gives the chance for new flows to begin the transmission. All these mechanisms ensure fast acceptance times of streaming flows in congestion. Moreover, it is possible to decrease the acceptance times of streaming flows without increasing the mean transmission times of elastic flows. It means that it is possible to improve the properties of admission decisions on streaming flows without significant degradation of transmission performance of elastic flows. The congestion control mechanisms have some drawbacks, i.e., transmission of elastic flows may be broken each time the flushing of the PFL content occurs. Moreover, the number of flows accepted in the AC block after flushing is, usually, too high.

In this paper we propose a new congestion control mechanism, called RPAEF (*Remove and Prioritize in access Active Elastic Flows*), which is based on assumptions more suitable for FAN than its predecessors and is more scalable.

The pseudo code for realizing the RPAEF functionality in FAN is presented in Tab. I. If a packet of a new flow arrives at the admission control block in congestion, the procedure of the RPAEF mechanism starts. The For loop is executed for each flow, which identifier j is written to the PFL (line 3). If an elastic flow with $ID = j$ is active for at least $active\_t$ [s], its ID is removed from the PFL and added to the PAFL (*Priority Access Flow List*) (lines 5-9, see also Fig. 1). A flow is selected as elastic if the number of enqueued bytes of its traffic is greater than or equal to the MTU (according to PFQ [2]). At the end of each RPAEF action, the value of $last\_RPAEF\_action$ is set to the current time (line 10). In congestion a packet of a flow is dropped (line 13). If packet p of new flow F arrives at the admission control block in the congestion-less state after a time given by the $prior\_access$ parameter from the last cleaning action on the PFL, the content of the PAFL is cleaned and packet p is accepted (lines 15-18 and line 25). If the PAFL is not empty, packet p is accepted if its flow ID is in the PAFL (lines 21-22). On the other hand,

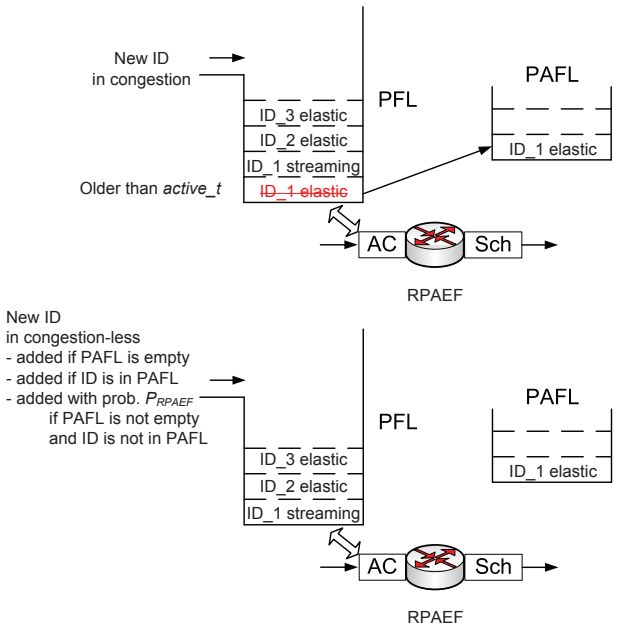

Fig. 1. The operation principle of RPAEF

TABLE I
PSEUDO CODE FOR REALIZING THE RPAEF FUNCTIONALITY IN FAN

```
1.  on arriving packet p of new flow in congestion
2.      current_t = Scheduler :: instance().clock()
3.      For (j = 1; j ≤ pfl_size; j++) do
4.      begin
5.          active_t(j) = current_t − first_operation_t(j)
6.          If flow_bytes(j) ≥ MTU && active_t(j) ≥ active_t then
7.          begin
8.              remove j from PFL
9.              add j to PAFL
10.             last_RPAEF_action = Scheduler :: instance().clock()
11.         end
12.     end
13.     drop p
14.  ****************** admission decision ********************
15.  on arriving packet p of new flow F in the congestion-less state
16.      current_t = Scheduler :: instance().clock()
17.      If current_t − last_RPAEF_action > prior_access then
18.          clean PAFL content
19.      If PAFL is not empty then
20.      begin
21.          If ID(F) is in the PAFL then
22.              accept packet p
23.          Else accept packet p with probability P_RPAEF
24.      end
25.      Else accept packet p
26.  ************************ end ***************************
```

when the PAFL is not empty and the flow ID is not in the PAFL, the packet p is accepted with probability P_{RPAEF} (line 23) (see Fig. 1).

The idea of this solution is to ensure a quick acceptance time of new streaming flows without breaks in transmission of elastic flows which identifiers are deleted from the PFL. The removed elastic flows are accepted again in the AC block immediately while the rest of flows begin transmission with low probability P_{RPAEF}. In such a case the UDP flows with small packets (streaming flows) have much more chances for

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

acceptance than the TCP flows with bigger packets. That is why the streaming flows has the precedence in acceptance over elastic ones. The RPAEF mechanism also allows for decreasing the total number of flows accepted after a cleaning action of the PFL content. This is a very important advantage of this algorithm. The implementation of the RPAEF mechanism in the cross-protect router is more complex than EFM or RAEF and the same as RBAEF, but does not increase the complexity and processing resources significantly.

A. Simulation analysis of FAN with the RPAEF mechanism

In this and following sections we present the results of carefully selected simulation experiments carried out in the ns-2 simulator. Basic simulation parameters are presented in each section. The detailed description of the simulation environment is given in [11].

The simulation analysis was provided to show the usefulness of the RPAEF algorithm. The mean acceptance time of new streaming flows and the mean number of elastic flows in the PFL were analyzed in the topology presented in Fig. 2.

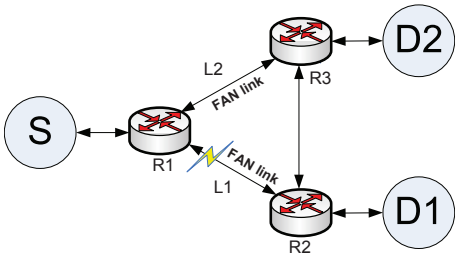

Fig. 2. The simulation topology

For simplicity, in our experiments we used one source node and two destination nodes. The S node is the source node of streaming and elastic flows, while the nodes $D1$ and $D2$ are destination nodes of traffic sent from the S node. It was assumed that bottleneck links L1 and L2 are FAN links of the 100 Mbit/s capacity each. The buffer was sized to 1000 packets which is a reasonable value for FAN links and the *MTU* was set to 1500 bytes. 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 and R2 while the traffic to node D2 is sent through nodes R1 and R3. By using such a topology it was assumed that link L2 was treated as a backup link with background traffic for packets sent normally through link L1. The effects of failures of link L1 at a chosen time instant were analyzed and presented in the experiments described below.

Firstly, 250 simulation runs (50 for basic FAN and 50 for each of four *active_time* values which were changed from 5 s to 20 s) were performed. The simulation duration was set to 500 s and at 200 s the L1 link was set down. The traffic pattern with Pareto distribution for calculating the volume of traffic to be sent by elastic flows was used. The number of background elastic flows was changed ranging from 400 to 1200. Half of them were destined to D1 and the rest to D2. The exponential distribution for generating the time intervals between beginnings of the transmissions of the elastic flows as well as for generating the start times of streaming flows was used. All of 40 streaming flows were destined to D1. The elastic traffic was treated as background traffic and used to saturate the analyzed links. We decided to analyze the VoIP connections realizing the Skype service. The packet size was set to 100 bytes and the transmission rate was set to 80 kbit/s for each of the streaming flows. We made our simulation runs in various conditions changing the number of selected parameters. We analyzed the acceptance time of each streaming flow in the AC block of router R1 for both outgoing links (L1 before failure and L2 after failure) and the number of accepted flows in L1 and L2 links before and after the L1 link failure, respectively. The measurement interval for the PL parameter was set to 50 ms while the FR values were estimated every 500 ms. The *max_priority_load* and the *min_fair_rate* 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 mean values of *waiting_time* (acceptance time of streaming flows) in both FAN links in function of the number of elastic flows active in background are presented in Fig. 3 and Fig. 4.

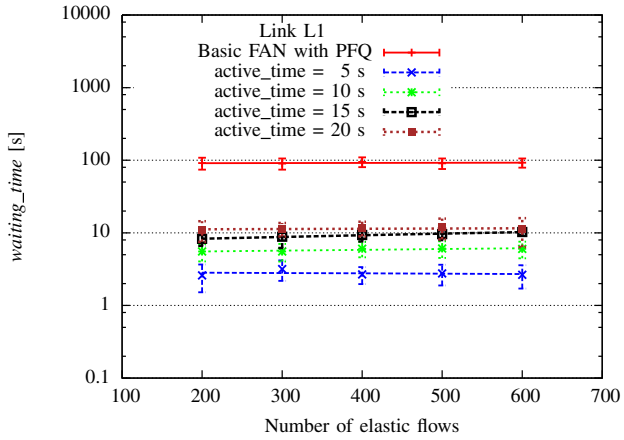

Fig. 3. The mean *waiting_time* in FAN link L1

As shown in Fig. 3 and Fig. 4 the values of *waiting_time* do not depend on the volume of background traffic in the examined range. It means that new streaming flows are accepted in the AC block independently of the number of elastic flows, which want to send the traffic. The *waiting_time* values increase alone with the *active_time* parameter, but are significantly smaller than in the basic FAN. It is assumed, according to [12], that setup time (post-selection delay) of local calls should be less than 6 s while for the international calls it should not exceed 11 s. When analyzing the L1 link we can see that in the examined range the best results were obtained when the *active_time* was set to 5 s. The *active_time*

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

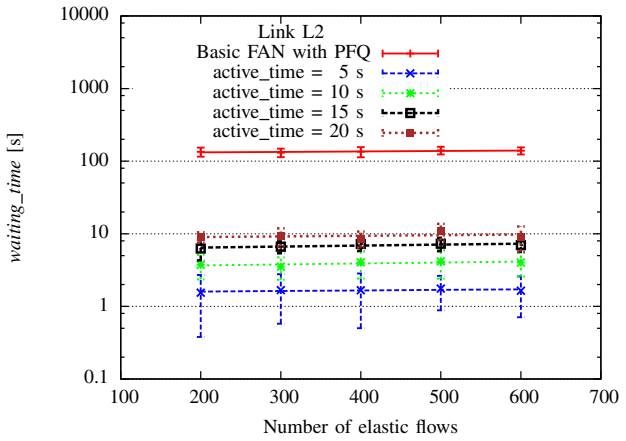

Fig. 4. The mean *waiting_time* in FAN link L2

equal to 10 s is also acceptable for local calls, while the values of 15 s and 20 s ensure sufficiently small *waiting_time* values for international VoIP connections. It is not desirable to set the *active_time* parameter to value lower than 5 s (the value for which the results are fully acceptable) because it unnecessarily increases the number of flushing actions. After redirecting the traffic from the basic route to the backup one the redirected flows have to compete with each other and with flows of the traffic normally sent through the L2 link. This is the reason why in the basic FAN the mean acceptance time of streaming flows in this link is longer than that observed for the L1 link. Moreover, even if we use the RPAEF mechanism the observed times for the L2 link are too long. The redirected streaming flows should be accepted in the backup link immediately.

The mean number of elastic flows in FAN with RPAEF accepted after any flushing action is presented in Fig. 5.

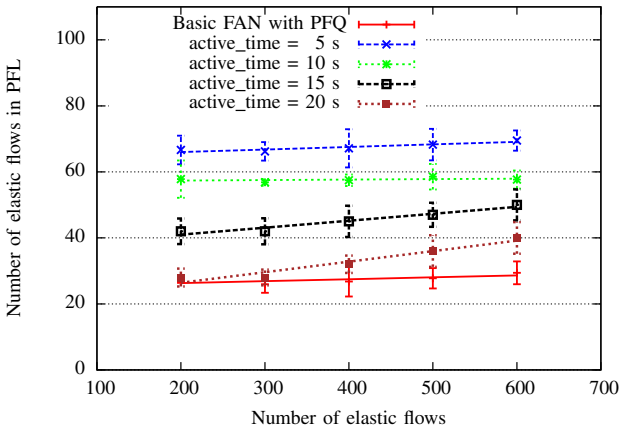

Fig. 5. The mean number of elastic flows in PFL in FAN with RPAEF

While the analysis of the *waiting_time* parameter gives similar results for all congestion control mechanisms proposed for FAN, the number of accepted elastic flows after flushing is smaller for the RPAEF algorithm than for the EFM, RAEF and RBAEF mechanisms (numerical results are not presented in the paper because of lack of space). It shows that RPAEF is more scalable. Unfortunately, as we can see, the obtained results are still not satisfactory. The mean number of accepted flows is too high when comparing it to the results presented for basic FAN. The limiting mechanism and the concept of implementing the GPFL presented in the following sections are proposed to deal with the RPAEF inconveniences. We can see in Fig. 5 that the mean number of accepted elastic flows after a flushing action is almost constant for low values of the *active_time* parameter. As the results for the *waiting_time* parameter are acceptable only for *active_time* equal to 5 s or 10 s, the further analysis is presented for these values and for FAN links with 200 elastic flows active in background in each of them.

IV. The Limiting Mechanism

In the RPAEF mechanism there are still too many flows accepted in the router after a flushing action, and in many cases it is difficult to ensure the proper transmission parameters for elastic flows (mean value of *fair_rate* should be close to *min_fair_rate*). In this section, we propose a mechanism which limits the number of flows accepted in the FAN routers. The main goal of the proposed algorithm is to ensure that in the time period between any two consecutive measurements of the *fair_rate* parameter, it is possible to add up to *N* identifiers to the PFL of the considered outgoing link. The *N* parameter is estimated from the following formula:

$$\begin{cases} N = 100/(min\_fair\_rate \times i) & \text{if } i > 0 \\ N = \infty & \text{if } i = 0 \end{cases} \quad (1)$$

where $i \in \mathbb{N}$ is the parameter which was changed in our simulation experiments to estimate the proper value of *N*. If i is set to 1 and the *min_fair_rate* is set to 5%, it means that up to 20 flows can be accepted in the router during one measurement period of *fair_rate*. We analyzed four cases, when i was set to 1, 2, 3 or 4.

TABLE II
Pseudo Code of Limiting Mechanism in FAN

```
1.  on a packet p of new flow F arrival in the congestion-less state
2.      If addmitted_flows_number > N then
3.          drop p
4.      Else
5.      begin
6.          addmitted_flows_number++
7.          add ID(F) to PFL
8.          proceed with p
9.      end
**************************************************************
10.     ****************** computing fair_rate ******************
11.     compute fair_rate
12.     addmitted_flows_number = 0
13.     ********************** end **************************
**************************************************************
```

The pseudo code of the limiting mechanism in FAN is presented in Tab. II. A new flow may be accepted at the admission control block in a congestion-less state only if the value of the $addmitted\_flows\_number$ parameter is lower than or equal

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

to N (lines 1-9). After accepting a new flow in the router, the $addmitted\_flows\_number$ value is incremented by 1 (line 6). Each time the measurement procedure of *fair_rate* is executed the $addmitted\_flows\_number$ parameter is set to zero (its initial value).

A. Simulation analysis of FAN with RPAEF and limiting mechanism

The simulation experiment was provided in the topology shown in Fig. 2. The number of elastic flows was set to 200 for each destination node. The analysis was provided for two values of the *active_time* parameter (5 s and 10 s). The rest of simulation parameters were set as in the previous case. The results are presented in Fig. 6 and Fig. 7.

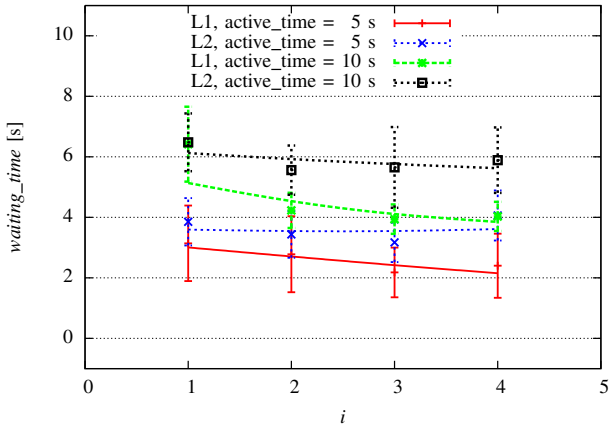

Fig. 6. The mean *waiting_time* in FAN link with RPAEF and the limiting mechanism

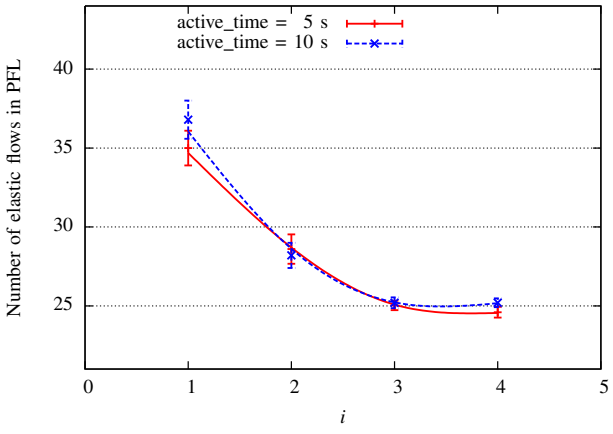

Fig. 7. The mean number of elastic flows in PFL in FAN with RPAEF and the limiting mechanism

In most cases, the values of *waiting_time* slightly decrease with increasing values of i in the examined range. The TCP flows, the packets of which are discarded, have to slow down their transmission rate. If i increases, fewer flows may be accepted during a *fair_rate* computation interval. It means that with increasing values of i more TCP flows are not accepted in the AC block and decrease their transmission windows. As an effect of it, the streaming flows (UDP flows) have chances to be accepted faster.

The main conclusion of the results presented in this section is that it is possible to decrease the number of elastic flows accepted in the AC block to the value observed for the basic FAN links (without congestion control mechanisms). This situation is observed for $i = 2$. For $i = 1$ the number of accepted flows is too high, while for the higher values of i, it is too low and not all rejected flows are accepted again after flushing. Moreover, for values of 5 s and 10 s of the the *active_time* parameter, the streaming flows begin their transmission in a short, acceptable time.

The limiting mechanism works properly in a network without failures. It is reasonable to accept the limited number of new flows when there is no need to redirect the traffic to the other route. When a network element fails, the traffic from the broken route is redirected to the alternative route. In such a case we should accept the IDs of all redirected streaming flows on the new route immediately. It means that in case of a network element failure, the limiting mechanism should work only for elastic flows. The possibility to implement such a solution gives the Global Protected Flow List (GPFL).

V. THE GPFL CONCEPT

The GPFL allows for continuous transmission of packets of streaming flows without breaks even when the traffic is rerouted after a link or node failure. The idea of using the GPFL is that streaming flows, which identifiers are written to it, are always accepted in FAN routers independently of the chosen route.

TABLE III
PSEUDO CODE FOR REALIZING THE GPFL CONCEPT IN FAN

```
1.  on a packet p of flow F arrival
2.      If ID(F) is in the PFL then
3.      begin
4.          If F is streaming then
5.              set F_prior = 1 in the GPFL
6.          Else set F_prior = 0 in the GPFL
7.          send p for queuing
8.      end
9.      Else (not in the PFL)
10.     begin
11.         If link is congested then
12.         begin
13.             If ID(F) is in GPFL and F_prior = 1 then
14.             begin
15.                 add ID(F) to PFL
16.                 send p for queuing
17.             end
18.             Else drop p
19.         end
20.         Else (link not congested)
21.         begin
22.             If ID(F) is not in GPFL then
23.                 add ID(F) to GPFL
24.             add ID(F) to PFL
25.             send p for queuing
26.         end
27.     end
```

978-1-4244-4148-8/09/$25.00 ©2009

The pseudo code for realizing the GPFL concept in FAN is provided in Tab. III. If a packet of a flow arrives at the admission control block and its ID is in the PFL of the outgoing link it means that its ID was previously added to the GPFL. Then, if the flow is of the streaming type, its priority is set to 1 in the GPFL (lines 1-5). For elastic flows the priority is set to 0 (line 6). On the other hand, if this flow was not previously accepted in the router and the link is congested, the flow's ID may be added to the PFL only if flow's ID is in the GPFL and its priority is set to 1 (lines 9-19). In a congestionless state, the ID of a new flow is added to the GPFL (if it is not there), and to the PFL, and the flow's packets may be sent (lines 20-27).

The implementation of the GPFL in FAN routers allows for slight modification of the limiting algorithm presented in Tab. II. In line 6 of this table we check if the identifier of a flow is in the GPFL and, if so, what is the value of its priority parameter. The $addmitted\_flows\_number$ variable should be increased only if the flow priority is set to zero. It ensures that all streaming flows from the redirected route are accepted in the new one and the number of other flows is limited.

A. Simulation analysis of FAN with RPAEF, limiting mechanism and GPFL

The simulation experiment presented in this section was provided in the same conditions as in the previous case but with the global list implemented. As was expected, the streaming flows are accepted immediately in the FAN link of the backup route (see Fig. 8, two lines overlaps one another). The mean number of elastic flows accepted after a flushing action is almost the same as in Fig. 7, thus, the revelant figure is not presented here. Summarizing the results, we can say that for *active_time* = 5 s or 10 s, $i = 2$, it is possible to assure fast acceptance of new streaming flows, good performance of transmission of elastic flows and proper network behavior after a link or node failure in FAN with RPAEF, limiting mechanism and GPFL.

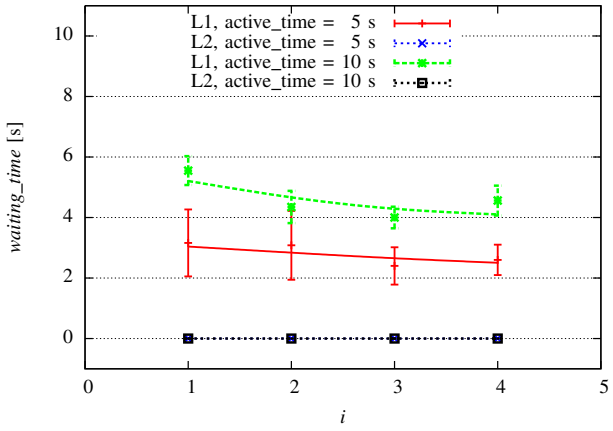

Fig. 8. The mean *waiting_time* in FAN link with RPAEF, limiting mechanism and GPFL

VI. Conclusion

The concept of a complete system for reliable transmission in Flow-Aware Networks is proposed and analyzed in the paper. FAN is a relatively new proposal to be used in new and existing networks to guarantee the QoS in an implicit way. This efficient and scalable solution meets the requirements of neutral networks and is considered as a very interesting and promising alternative to currently used QoS architectures. Many researchers work on FAN all over the world to enhance its functionality.

The RPAEF mechanism is the most promising solution of all congestion control mechanisms proposed for FAN. It ensures fast acceptance times of streaming flows and good transmission performance for elastic flows. The RPAEF algorithm improved by implementing the limiting mechanism and GPFL creates a scalable system for reliable transmission in FAN. By implementing and using it, we prioritize the streaming flows in a real way. Such flows are accepted quickly and they send their traffic without breaks even after a network element failure. The outages in transmission of elastic flows are observed only after failure. In normal conditions, once accepted, the flows are protected.

The system presented and analyzed in the paper is promising and may be used in the future Internet.

Acknowledgements

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] J. Roberts and S. S. Oueslati, "Quality of Service by Flow Aware Networking," *Philosophical Transactions of The Royal Society of London*, vol. 358, pp. 2197–2207, September 2000.

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

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.

[4] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture an Overview," IETF RFC 1633, June 1994.

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

[6] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts, "On the scalability of fair queueing," in *ACM HotNets-III*, San Diego, USA, November 2004.

[7] J. Joung, J. Song, and S. S. Lee, "Flow-Based QoS Management Architectures for the Next Generation Network," *ETRI Journal*, vol. 30, pp. 238–248, April 2008.

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

[9] ——, "The Impact of Congestion Control Mechanisms for Flow-Aware Networks on Traffic Assignment in Two Router Architectures," in *ICLAN 2008*, Toulouse, France, December 2008.

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

[11] http://www.kt.agh.edu.pl/∼jdomzal/sim_param_globecom09.pdf.

[12] ITU-T, "Network grade of service parameters and target values for circuit-switched services in the evolving ISDN," Recommendation ITU-T E.721, May 1999.

978-1-4244-4148-8/09/$25.00 ©2009