Per User Fairness in Flow-Aware Networks

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

***Abstract*—A new fairness concept for Flow-Aware Networks (FAN) is presented and analyzed in the paper. In contrast to basic FAN, where elastic flows are treated similarly, it assumes fair access to the resources for each user independently of how many flows it generates. A new method for estimating the values of the *fair_rate*, which is the key congestion control parameter in FAN is also provided. The new solution allows for reducing the oscillations of the periodically measured values which influences more stable transmission in the network. The new method allows for describing the traffic in a more realistic way, simultaneously simplifying its implementation in the FAN routers. Finally, the results of the simulation analysis of FAN with the new fairness algorithm and the Remove and Accept Most Active Flows (RAMAF) congestion control mechanism are presented for two versions of the TCP protocol: NewReno and NewJersey.**

***Index Terms*—fairness; Flow-Aware Networks; Quality of Service; congestion control; wireless transmission; TCP**

I. Introduction

The Internet traffic grows rapidly every year. New applications and services consume more and more bandwidth. Therefore Internet Service Providers (ISPs) as well as large carriers have to constantly upgrade their networks.

It is a real challenge for network operators to transmit traffic with guaranteed quality. Today, they prefer to add extra bandwidth rather than to implement usually complicated Quality of Service (QoS) architectures. However, in such cases traffic is not differentiated and sent in the best effort mode. Sometimes it may be blocked or delayed.

In this paper, we propose a new approach to fairness assurance in Flow-Aware Networks (FAN) [1]. In FAN, traffic is sent by streaming or elastic flows and classified based on the transmission rates. The values of two observed parameters (*fair_rate* and *priority_load*) decide whether new flows can be accepted in a router or not. FAN is a scalable architecture, which ensures that packets of streaming flows are sent first (with high priority). On the other hand, the packets of elastic flows are served realizing the best effort service. One of the most important advantages of FAN is that flows are sent fairly. It means that, for similar traffic, bandwidth is divided equally for each flow.

Fairness may be perceived in many ways. For example, best effort transmission is fair because each packet has the same chance for transmission. In FAN, each flow is assigned the same rate, which is also fair. However, in current networks we need more sophisticated fairness. The multi-flow applications (e.g., P2P) may overload access networks and cause that bandwidth per flow is very low. Such P2P flows may consume almost all bandwidth and the quality of transmission of the other traffic may be drastically deteriorated. This brief discussion shows that fairness per flow is not the best solution. In this paper, we present a new concept of fairness for FAN. In our proposal, we assume that similar end users get equal bandwidth in the outgoing link. In opposition to the original solution proposed for FAN, where fair bandwidth is ensured for each flow, it realizes the concept: equal bandwidth for end users.

The rest of the paper is organized as follows. Section II presents the main assumptions and a brief description of the FAN architecture. In Section III, we present the *fair_rate* congestion control parameter used in FAN and propose the new method for calculating its values. Section IV presents the new concept for realizing fairness in FAN. In Section V, the results of carefully selected simulation experiments are presented. The key point of this section is to show the effectiveness of transmission in a FAN network with the new fairness algorithm. Section VI concludes the paper.

II. Flow-Aware Networks

The concept of FAN was proposed as an answer to inconveniences of currently used QoS architectures. FAN ensures stable packet transmission using only the minimal information from the network. It is possible because of implementing the new cross-protect router architecture, which is presented in Fig. 1. Two main blocks are specified in the FAN router: the Measurement Based Admission Control (MBAC) and the scheduler [2]. The former decides of accepting or dropping the incoming packets, while the latter schedules the accepted packets in the queues. Both parts of the cross-protect router depend on each other. If an identifier (ID) of the flow represented by an incoming packet is on the Protected Flow List (PFL), the packet is accepted and sent to the scheduler block. On the other hand, in the congestion-less state, the ID of the flow is written to the PFL and the packet is accepted for queuing. If the ID is not on the PFL and the outgoing link is congested, the incoming packet must be dropped. The periodical measurements of two congestion control parameters, the *fair_rate* and the *priority_load* are provided in the scheduler block. If the *fair_rate* value is lower than the *min_fair_rate* (minimum allowed value of the *fair_rate*) or the *priority_load* value is higher than the *max_priority_load* (maximum allowed value of the *priority_load*) the congestion is noticed. There are

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

two types of flows in FAN: elastic (usually realize the best effort transmission) and streaming (low rate flows, e.g., VoIP connections).

There are two well known FAN architectures. They use different scheduling mechanisms. In the first version, the Priority Fair Queuing (PFQ) algorithm is used for scheduling the packets. It is based on the Start-time Fair Queuing (SFQ) algorithm and inherits the advantages from it through the possibility of prioritizing the selected packets in the scheduler module. Priority Deficit Round Robin (PDRR), used in the second version of FAN, is a fair queuing algorithm based on the Deficit Round Robin (DRR) scheduling mechanism. PDRR inherits the advantages from DRR (e.g., $O(1)$ complexity and fairness) and is enriched by packet prioritizing possibilities in the scheduler block. The full description of PFQ and PDRR with the algorithms for packet queuing and scheduling are presented in [3] and [4]. There is also the third FAN architecture, called Approximate Flow-Aware Networking (AFAN) [5], which is based on the Approximate Fair Dropping (AFD) algorithm for packet scheduling [6].

III. New method for estimating the fair_rate

In the scheduler, two parameters are measured:

- *fair_rate* — the maximum rate that is or might be realized by a flow,
- *priority_load* — the quotient, which represents the rate of incoming priority packets with reference to the link capacity.

Both parameters are used to arbitrate if the outgoing link is congested or not. In the absolute majority of cases, the *fair_rate* is the parameter which decides on congestion. Moreover, it has an important impact on transmission in the FAN link.

In this section, we propose a new method for estimating the values of the *fair_rate* parameter in each FAN architecture.

In PFQ, the *fair_rate* is computed from the following formula:

$$fair\_rate = \frac{max\{S \times C, (vt(t_2) - vt(t_1)) \times 8\}}{t_2 - t_1} \quad (1)$$

where *vt*(*t*) is the *virtual_time* in time *t* and represents the start tag of the last packet of the fictitious permanently backlogged flow, which sends 1 byte long packets between packets of real flows in proper order (compatible with the algorithm assumptions), (t_1, t_2) is the time period measured in seconds, *S* is the total length of inactivity in the transmission during the (t_1, t_2) period, *C* is the link bit rate.

In PDRR, the *fair_rate* is computed from the following formula:

$$fair\_rate = \frac{max\{S \times C, fair\_bytes \times 8\}}{t_2 - t_1} \quad (2)$$

where *fair_bytes* is a number of bytes, which could be sent by a fictitious permanently backlogged flow during the time interval (t_1, t_2), *S* is the total length of inactivity in the transmission during the (t_1, t_2) period, *C* is the link bit rate.

The smoothing parameter α is applied in both versions, such that:

$$\begin{aligned} fair\_rate(n) &= \alpha \times fair\_rate(n-1) \\ &+(1-\alpha) \times measured\_fair\_rate(n) \end{aligned} \quad (3)$$

where $fair\_rate(n)$ is the value of *fair_rate* in the n-th iteration and the $measured\_fair\_rate$ is the value calculated from the formula (1) or (2) in the n-th iteration.

The methods of estimating the *fair_rate* have some drawbacks and can be improved. Here, we propose a new method for calculating the *fair_rate*, which may be used in each FAN architecture.

Adding a fictitious flow to estimate the *fair_rate*, as in the known solutions, increases the complexity of the cross-protect router. We suggests to calculate the *fair_rate*, for each FAN version, in a simpler way from the following formula:

$$fair\_rate = \frac{max\{S \times C, FB \times 8\}}{t_2 - t_1} \quad (4)$$

where FB is the number of bytes sent by elastic flows (not the number of bytes which could be sent by a fictitious flow) during the time interval (t_1, t_2), measured in seconds, divided by the number of elastic flows, which identifiers are written to the PFL, *S* is the total length of inactivity in the transmission during the measurement period, *C* is the link bit rate. The flow is classified as elastic if its enqueued number of bytes is greater than the maximum allowed transfer unit (in bytes). It is easy to count the number of bytes transmitted by elastic flows during the measuring time period and to get the number of elastic flows from the PFL. The *fair_rate* calculated in this way does not represent the real rate realized by elastic flows but estimate it with precision sufficient to decide on congestion. Our solution is simpler than the original solutions. It is easier to count the bytes sent by source nodes than to estimate the number of bytes which may be sent by a fictitious flow.

IV. New fairness concept for FAN — equal bandwidth for end users

In this section, we propose and analyze a new fairness algorithm for FAN, which realizes the equal bandwidth for each user concept. There are many solutions which allow for fair packet scheduling in current networks, e.g., Weighted Fair Queuing (WFQ), SFQ or DRR. There are also many proposals for admission control, e.g., measurement based admission control (MBAC) or parameter based admission control (PBAC). In this paper, we propose a new algorithm, which combines WFQ and MBAC. Its role is to ensure fair access to XP routers for each user. Currently, in FAN, new flows can begin transmission only in the congestion-less state. Moreover, each accepted elastic flow sends its traffic as a best effort service and may transmit it with the same rate as other flows. Such an approach is called fairness per flow. There are some drawbacks of this solution. Firstly, elastic flows may occupy the link for a long time, significantly increasing the acceptance times of

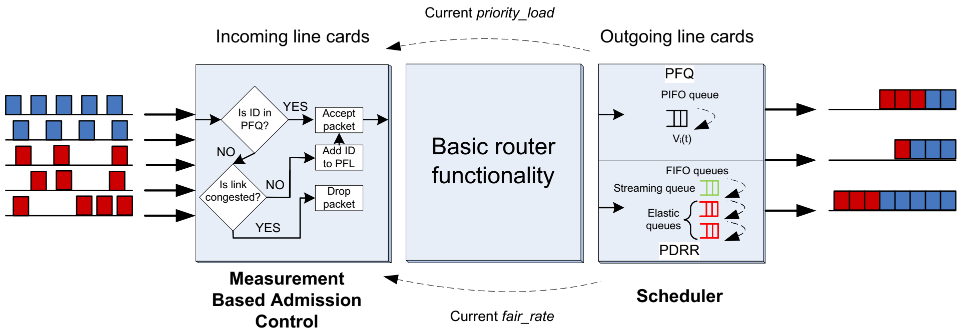

Fig. 1. The cross-protect router architecture

streaming flows. Secondly, some end users may generate many flows at the same time (e.g., by a P2P application) and cause that the outgoing link is congested. In such a case, the other end users may not be able to begin any transmission.

The solution to the first problem may be the Remove and Accept Most Active Flows (RAMAF) congestion control mechanism proposed and analyzed in [7]. RAMAF periodically removes IDs of a number of most active flows from the PFL and writes them to the Priority in Access Flow List (PAFL). This number is set dynamically based on the queue occupancy. Next, in a congestion-less state, the removed IDs are moved from the PAFL to the PFL again. If new elastic flows are accepted in the router after a clearing action of the PFL, they are removed from the list immediately. The RAMAF mechanism allows for dynamic changes of values of the *fair_rate* parameter around the minimum acceptable value, and in consequence gives chances to new flows to begin their transmission. The streaming flows, which usually realize voice or video transmission should be accepted in the router as soon as possible. The acceptable time for international calls should not be greater than 11 seconds for the 95% of the calls, while for the local calls it should not exceed 6 seconds [8]. The simulation analysis provided in [7] confirms that such a mechanism ensures reasonable acceptance times of streaming flows in the cross-protect router without significant decrease of transmission quality of elastic flows.

The second problem may be solved by limiting the number of flows accepted during a *fair_rate* measurement interval. In such a solution, we have to fix the number of flows generated by each source node, which may be accepted in the router during a *fair_rate* measurement interval. The motivation for such a proposal is to ensure fair access to the resources for each active source node. The pseudo code of the mechanism is presented in Tab. I (the code for the similar solution was presented in [9]). 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 (set for each source node) is lower than or equal to $N(i)$ (lines 1-9). $N(i)$ is fixed based on weights set to the source nodes. After accepting a new flow in the router, the $addmitted\_flows\_number$ value for the particular source node is incremented by 1 (line 6). Each time the measurement procedure of *fair_rate* is executed the $addmitted\_flows\_number$ parameter for each source node is set to zero (its initial value).

The maximum number of flows which may be accepted during one measurement period of *fair_rate* should be fixed for each router. In our simulation experiments, for simplicity, we decided that every *fair_rate* measurement interval we could accept maximum one flow generated by a source node.

TABLE I
PSEUDO CODE OF LIMITING MECHANISM IN FAN

1. on a packet p of new flow F arrival in the congestion-less state generated by source S_i
2. **If** $addmitted\_flows\_number(i) > N(i)$ **then**
3. drop p
4. **Else**
5. begin
6. $addmitted\_flows\_number(i)$++
7. add ID(F) to PFL
8. proceed with p
9. end

****

10. ****************** computing *fair_rate* *******************
11. compute fair_rate
12. **For** $(i = 0; i \leq source\_number; i{+}{+})\{$
13. $addmitted\_flows\_number(i) = 0$
14. $\}$
15. ************************ end *************************

****

V. SIMULATION ANALYSIS

In this section, we present the results of carefully selected simulation experiments run in the ns-2 simulator. Firstly, we present the results of the simulation analysis of the new method for calculating the *fair_rate* values provided for the FAN architecture with the PFQ or PDRR algorithm. The results for the AFAN architecture are similar to those presented for FAN with PDRR and therefore not shown here. Secondly, we present the results of the simulation experiments of FAN with the new fairness algorithm and the RAMAF mechanism.

A. Simulation analysis of the new method for estimating the fair_rate

In this part, we made 240 simulation runs (60 for each FAN version with a different method for calculating the *fair_rate*) in various conditions to show the mean deviation from the *min_fair_rate* value in function of the smoothing parameter

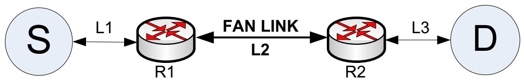

Fig. 2. Simulation topology

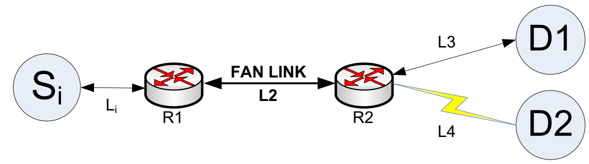

Fig. 4. Simulation topology with wireless link

(see Fig. (3)). The simulation topology is presented in Fig. 2. The traffic is sent from the source node S through the FAN link to the destination node D. The simulated topology is very simple, yet adequate to analyze the new method for estimating the *fair_rate* values in the FAN networks. The reason behind such a statement is that all nodes in FAN operate independently and all the decisions are taken without any information from the network. Therefore, the topology is sufficient to demonstrate the operation of the analyzed algorithm.

We provided the traffic pattern with Pareto distribution for calculating the volume of traffic to be sent by each of 200 elastic flows in the FAN link with capacity equal to 100 Mb/s. We used the exponential distribution for generating the time intervals between beginnings of the transmissions of the flows (the mean value of the inter-arrival time was set to 0.1 s). The duration of each simulation run was set to 1200 s. The measurement interval for the *priority_load* parameter was set to 50 ms while the *fair_rate* values were estimated every 0.5 s. These values were chosen experimentally to guarantee the stable transmission. The *max_priority_load* parameter was set to 70% of link capacity, the *min_fair_rate* parameter was set to 5% of link capacity and the warm-up period was set to 20 s. 95% confidence intervals were calculated by using the Student's t-distribution.

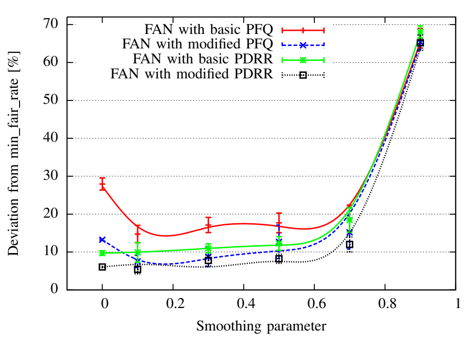

Fig. 3. The mean deviation from *min_fair_rate*

The values of *fair_rate* change in time and oscillate around *min_fair_rate*. The situation is better if the deviations from *min_fair_rate* are lower. We can see that for FAN with PDRR, the analyzed mean deviation is lower than for FAN with PFQ. The new method for estimating the *fair_rate* parameter allows for decreasing the mean values of deviations in both FAN versions. The gain is from several to a dozen or so percent. It means that the solution proposed by us ensures more stable transmission in FAN. The mean values of the *fair_rate* may be approximately calculated as the difference between the *min_fair_rate* and the mean deviation of *fair_rate*.

The values of the mean deviation from the *min_fair_rate* change in function of the smoothing parameter. We can observe that, for FAN with PFQ, the values of the estimated deviation are most suitable in the range from 0.1 to 0.6 of the values of the smoothing parameter. In this range, the value of the deviation from *min_fair_rate* increases insignificantly. For FAN with PDRR the appropriate range of the smoothing parameter is even wider. In this FAN version, it is not necessary to use the smoothing parameter. In FAN with PFQ the best results are observed for relatively small values of the smoothing parameter.

B. Simulation analysis of FAN with new fairness algorithm

In the experiments described in this section we used the topology presented in Fig. 4. In comparison to the previous experiments, we added the wireless link (L4). The goal is to show the impact of the new fairness algorithm and the RAMAF mechanism on transmission in FAN with end users connected by wireless links. Moreover, we have analyzed two cases with different versions of the TCP protocol. The TCP NewReno is well known. The TCP NewJersey, described in details in [10] and [11], allows for differentiation what cause the packet losses in the wireless link. If a link on a packet route is congested then the transmitted packet is marked with the Congestion Warning (CW) bit. If the ACK of a packet arrives at the sender without the CW mark, it proceeds as NewReno. The rate control procedure is activated if the source node receives the ACK or third duplicate ACK (DUPACK) marked with the CW bit. Then, the window size is adjusted for further transmission. It is assumed that a packet drop is caused by the random error when the sender receives the third DUPACK without the CW mark. In FAN, packets are marked in congestion but only if the PAFL is empty. If the packets are dropped due to the wireless link errors, then the fast retransmit procedure is called without adjusting the window size. It allows for better bandwidth utilization, in comparison to, e.g., TCP NewReno, when after each retransmission of a packet the window size is decreased. We have observed the goodput (the number of packets received by end users) of the tested elastic flow during the transmission from the source node S_4 to the destination node $D2$.

The four source nodes S_i (i=1,2,3,4) were connected to the FAN router R1 via links L$_i$ (each with 1 Gbit/s capacity). The FAN routers (R1, R2) were connected via link L2 (with 100 Mbit/s capacity). The R2 router was connected to the destination node $D1$ via link L3 (with 1 Gbit/s capacity) and to the destination node $D2$ by a wireless link L4 (with 5 Mbit/s capacity).

We analyzed cases with 400 elastic flows (FTP connections) and 20 streaming flows (VoIP connections via Skype service) transmitting their traffic from the source nodes to the destination node $D1$. The traffic pattern followed the Pareto distribution, and thus the volume of traffic to be sent by the elastic flows could be calculated accordingly. 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 adopted. We analyzed two cases, which we called ”low-loaded FAN link” and ”high-loaded FAN link”. In both cases S_1 generated 160 elastic flows, S_2 80, S_3 40 and S_4 20. In the first case, S_1 generated flows with the mean value of the interarrival time set to 0.4 s. For S_2, S_3 and S_4 these values were set to 0.3 s, 0.2 s and 0.1 s, respectively. In the second case, the mean values of the exponential distribution were set to 0.025 s, 0.05 s, 0.075 s and 0.1 s for the appropriate source nodes. Such set of parameters ensured that in the second case the FAN link was more heavily loaded. The packet size for the elastic flows was set to be 1000 bytes, while that for the streaming flows was set to be 100 bytes. The transmission rate of streaming flows was set to be 80 kbit/s. We also added one elastic flow which began its transmission when the simulation started and was sending its traffic during the whole simulation from the source node S_4 to the destination node D2. For this flow, we observed the goodput. The duration of each simulation run was set to 300 s, which allowed to observe the acceptance times of streaming flows (*waiting_time*) in the R1 router and goodput of elastic flows. We assumed that the random link error rate at the wireless bottleneck link varied from 0.01% (0.0001 in the following figures) to 10% (0.1 in the following figures). The smoothing parameter for calculating the *fair_rate* values was set to 0.1 (as a result of observations presented in the previous section). The rest of the simulation parameters were set as in the previous experiments.

The efficiency of the wireless link (goodput / wireless link capacity) versus the packet loss rate for the FAN with TCP NewReno is presented in Fig. 5 and Fig. 6. The results for FAN with TCP NewJersey are shown in Fig. 7 and Fig. 8. The phrase fair FAN is used as a short description of FAN with the new limiting algorithm.

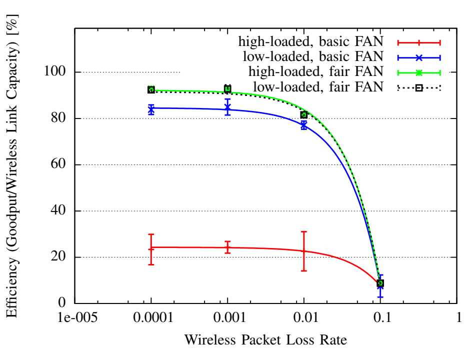

Fig. 5. TCP efficiency in FAN with TCP NewReno

Firstly, we have to note, that for low values of the wireless packet loss, the results are slightly better for TCP NewReno. On the other hand, when the values of the wireless packet loss increase, TCP NewJersey ensures better transmission effectiveness (for the random link error equal to 10% the gain is a dozen or so percent). It is a very important advantage of TCP NewJersey. Usually, wireless links generate some errors and many packets are dropped in such links. The number of errors grows up significantly if the distance between an end user and the access point increases.

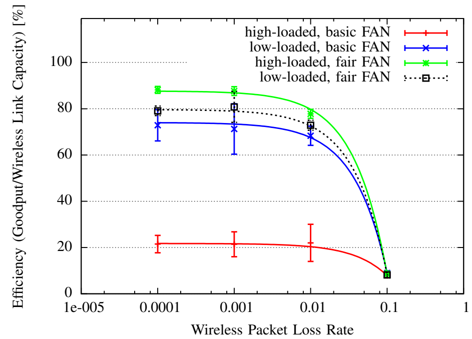

Fig. 6. TCP efficiency in fair FAN with RAMAF and TCP NewReno

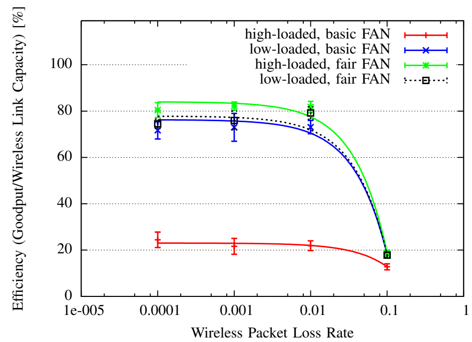

Fig. 7. TCP efficiency in FAN with TCP NewJersey

Secondly, our analysis proves that the RAMAF mechanism ensures short and acceptable values of *waiting_times* (acceptance times of streaming flows in router R1). The results presented in Tab. II show the strong advantages of the RAMAF mechanism. By using it, we can reduce acceptance times for streaming flows to several seconds, which is very important e.g., for VoIP connections. Moreover, the efficiency of transmission of elastic flows is reduced insignificantly.

The new algorithm for estimating the *fair_rate* values is less complex in comparison to the original solution and ensures lower deviations around the *min_fair_rate* in the analyzed cases what guaranties better performance and more stable transmission in the link. By more stable transmission we mean the case where the transmission rates of the elastic flows change in time less dynamically.

At the end, we have to note that the limiting mechanism proposed by us in this paper ensures fairness per user even if the outgoing link is highly loaded. For the low-loaded link, the difference between the obtained results is not so big but worth

TABLE II
THE WAITING_TIME VALUES IN FAN WITH WIRELESS LINK

TCP vers.	**Pkt Loss Rate**	***waiting_time* [s]** (low-loaded link)	***waiting_time* [s]** (high-loaded link)	***waiting_time* [s]** (low-loaded link)	***waiting_time* [s]** (high-loaded link)
		Basic FAN		RAMAF	
NewReno (FAN)	0.0001	16.91±6.47	>250	3.61±1.55	1.88±0.58
	0.001	21.51±5.64	>250	3.61±1.61	2.85±1.59
	0.01	27.30±4.34	>250	2.81±2.00	3.51±1.32
	0.1	23.61±9.79	>250	4.11±1.02	1.91±0.81
NewReno (fair FAN)	0.0001	7.90±3.15	8.10±4.08	3.41±0.52	3.91±0.28
	0.001	7.20±3.90	7.60±3.02	3.81±0.71	2.13±1.68
	0.01	6.91±2.54	9.11±4.35	2.78±1.60	2.98±1.79
	0.1	6.61±3.24	7.81±2.22	3.18±1.79	3.01±1.75
NewJersey (FAN)	0.0001	45.93±13.44	>250	2.64±1.38	2.02±0.87
	0.001	53.30±12.79	>250	3.31±0.94	2.56±1.09
	0.01	58.61±20.00	>250	3.06±1.61	3.45±1.81
	0.1	40.81±15.06	>250	3.39±1.65	2.38±1.75
NewJersey (fair FAN)	0.0001	27.31±6.80	22.91±8.09	3.21±1.29	2.26±1.12
	0.001	32.51±3.77	10.41±5.00	3.47±1.76	2.71±0.71
	0.01	31.31±7.56	19.10±8.27	2.61±0.28	2.31±1.04
	0.1	29.00±3.37	21.51±15.06	3.61±0.52	2.91±1.67

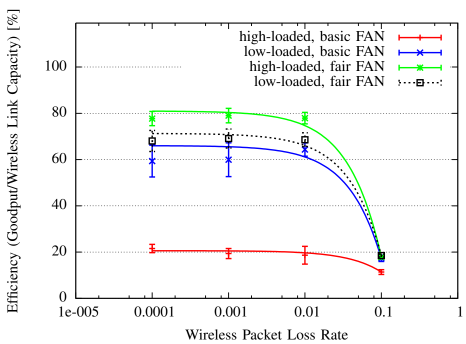

Fig. 8. TCP efficiency in fair FAN with RAMAF and TCP NewJersey

noting in both analyzed cases (basic FAN and fair FAN). The efficiency of transmission in basic FAN significantly decreases if sources begin to transmit more flows (in fact flows were generated more frequently). In the case where our algorithm works in a high-loaded link, the efficiency of transmission is similar or even slightly better to that observed for a low-loaded link. It means that independently of volume of traffic generated by the source nodes the fairness per user is achieved.

VI. CONCLUSION

Flow-Aware Networking is a promising concept for using in the Future Internet. It is simple and allows for ensuring the QoS for implicitly selected traffic. However, FAN is a relatively new proposal and still needs some improvements.

The new method for estimating the *fair_rate* in the scheduling block of the cross-protect router, proposed and described in this paper, allows for a more stable transmission in FAN.

The limiting algorithm proposed in the paper ensures fairness per user. Such a concept is desirable or even necessary, especially in cases where some source nodes generate many flows. Without the algorithm, traffic generated by some source nodes may occupy the outgoing link for a long time giving no chances for the other users to begin their transmission.

The simulation analysis presented in the paper shows that the implementation of the new algorithm for estimating the *fair_rate* values along with the RAMAF mechanism and the limiting algorithm, which decides on fair access of each user to the resources ensures fair, stable and effective transmission of elastic flows and short acceptance times of streaming flows. Moreover, the use of the TCP NewJersey ensures more effective transmission of elastic flows in wireless links.

ACKNOWLEDGEMENTS

The research was carried out with the support of the project ”Implementation of Cross-Protect router” founded by the National Centre for Research and Development under the LIDER programme.

REFERENCES

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

[2] ——, “Method and a device for implicit differentiation of quality of service in a network,” *United States Patent 2004/0213265 A1*, October 2004.

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

[4] ——, “Implicit Service Differentiation using Deficit Round Robin,” in *ITC19*, Beijing, China, August/September 2005.

[5] J. Domzal and A. Jajszczyk, “Approximate Flow-Aware Networking,” in *IEEE ICC 2009*, Dresden, Germany, June 2009.

[6] R. P. K. Psounis and B. Prabhakar, “Approximate Fair Dropping for Variable-Length Packets,” *Micro, IEEE*, vol. 21, pp. 48–56, January 2001.

[7] J. Domzal, N. Ansari, and A. Jajszczyk, “Congestion Control in Wireless Flow-Aware Networks,” in *IEEE ICC 2011*, Kyoto, Japan, June 2011 (to be published, available at www.kt.agh.edu.pl/∼jdomzal/ICC11.pdf).

[8] “Network grade of service parameters and target values for circuit-switched services in the evolving ISDN,” Rec. ITU-T E.721, May 1999.

[9] J. Domzal, R. Wojcik, and A. Jajszczyk, “Reliable Transmission in Flow-Aware Networks,” in *IEEE Globecom 2009*, Honolulu, USA, November-December 2009.

[10] K. Xu, Y. Tian, and N. Ansari, “Improving TCP Performance in Integrated Wireless Communications Networks,” *Computer Networks*, vol. 47, pp. 219–237, February 2005.

[11] ——, “TCP-Jersey for Wireless IP Communications,” *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 747–756, May 2004.