Intelligent Routing in Congested Approximate Flow-Aware Networks

Jerzy Domżał, *IEEE Member*

***Abstract*—A new algorithm for intelligent routing in Approximate Flow-Aware Networks (AFAN) is presented and analyzed in the paper. In contrast to basic AFAN, where cross-protect routers block excessive traffic under congestion, it assumes a possibility of using other routes to destination nodes based on the idea similar to load balancing. The analysis is provided for the AFAN concept, which is the newest version of Flow-Aware Networks (FAN). The simulation experiments, discussed in the paper, were provided for basic AFAN and for the network with one of two congestion control mechanisms, the Remove and Accept Most Active Flows (RAMAF) and Simple Congestion Control Mechanism (SCCM). These solutions allow for fast acceptance of streaming flows independently of the volume of traffic load in a network. The results are presented for four different simulation topologies and prove the advantages of the proposed intelligent routing approach. Moreover, it is shown that the SCCM ensures better transmission properties than the RAMAF.**

***Index Terms*—Flow-Aware Networks; Approximate Flow-Aware Networks; Quality of Service; congestion control; routing**

I. Introduction

Internet is changing. This sentence has been true since the first messages were sent between two nodes years ago. There are some milestones, which may be easily identified over the last fifteen years. Among them are web pages which definitely caused that Internet became popular all over the world. The fast progress in optical networks allowed for faster transmission with better quality. P2P networks opened a door for sharing resources between users. Finally, what we observe now, streaming transmissions changed computers in a user-friendly multi-functional machines.

Streaming transmissions supporting web services generate now the majority of traffic in the Internet [1]. It is a real challenge for network operators to ensure a proper quality of service (QoS) for such traffic. In many cases, not the transmission rate is the key point for transmission of streaming flows. The most important factors are the call acceptance delay of a connection, transmission delay, packet losses and connection reliability. To be up to these requirements, network operators usually add extra bandwidth rather than implement complicated QoS architectures.

In this paper, the concept of Approximate Flow-Aware Networks (AFAN) is presented as an architecture for Future Internet. It was first presented in [2] and assumes that streaming flows, e.g., VoIP or video streaming connections are transmitted with a high priority in a network. In AFAN, the packets are scheduled based on the Approximate Fair Dropping (AFD) algorithm and as a result the packet service is less complicated than in the previous solutions [3]. Moreover, the additional improvement of streaming packets service is ensured by the Remove and Accept Most Active Flows (RAMAF) congestion control mechanism introduced in [4]. The goal of it is to allow for fast acceptance of streaming flows in FAN routers even in congestion. In this paper, the new congestion control mechanism, called Simple Congestion Control Mechanism (SCCM) is proposed and compared with RAMAF.

The main aim of the paper is, however, to present the new algorithm for intelligent routing in AFAN. Without congestion control mechanisms, new flows cannot begin transmission if the outgoing link is congested. While there is no intelligence according to the routing protocols in the AFAN routers, the route between source and destination nodes is always chosen through the same path. However, there may exist many paths between such nodes and in congestion they should be used. The algorithm proposed in this paper meets the requirements presented above and may be implemented to improve network performance.

The rest of the paper is organized as follows. Section II presents the main assumptions and a brief description of the AFAN architecture. In Section III, the new algorithm for intelligent routing in Approximate Flow-Aware Networks is described. Sections IV and V present the congestion control mechanisms for AFAN, the RAMAF and SCCM, respectively. In Section VI, the results of carefully selected simulation experiments are discussed. The key point of this section is to show the volume of traffic which may be sent in a network when the new solutions proposed in the paper are implemented. Section VII concludes the paper.

II. Approximate Flow-Aware Networks

Approximate Flow-Aware Networks were proposed in 2009 in [2] as a new version of Flow-Aware Networks (FAN). The Flow-Aware Networking concept was introduced by J. Roberts and S. Oueslati in [5] and, then, presented as a complete system in 2004 [6]. The main assumption of FAN is that traffic is sent by flows. Two traffic types are considered: elastic and streaming. The flows which belong to the first group realize the best effort transmission. These are bandwidth consuming flows, that usually transmit data traffic, e.g., ftp or P2P connections. The second group encompasses the flows with limited rate but sensitive to delays and drops, e.g., VoIP or video streaming connections.

J. Domżał is with the Department of Telecommunications, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland (e-mail: domzal@kt.agh.edu.pl).

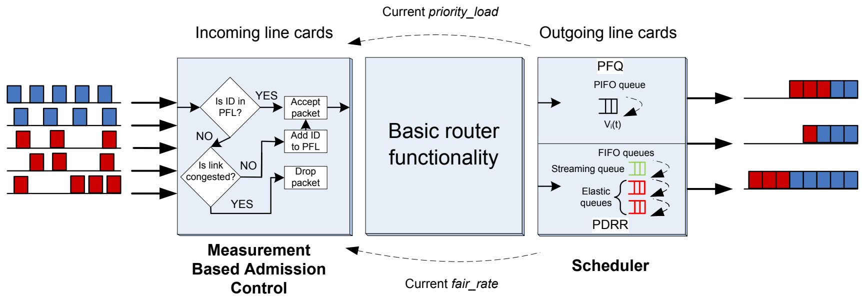

Fig. 1. The cross-protect router architecture

The main element of FAN architecture is the cross-protect router (also called XP router) which is presented in Fig. 1. In the Measurement Based Admission Control (MBAC) block packets are implicitly classified and decisions of accepting or dropping them are taken. If the outgoing link is not congested all incoming packets are accepted and identifiers of flows represented by these packets are added to the Protected Flow List (PFL). On the other hand, in congestion, only packets of flows which identifiers are in the PFL are accepted. In the scheduler block the values of two parameters are periodically calculated:

- *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.

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.

The second main role of the scheduling block is to schedule the accepted packets in a proper way. The authors of FAN proposed two scheduling algorithms which determine two possible versions of FAN implementation. In the first proposal, the Priority Fair Queuing (PFQ) algorithm is used. 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), proposed for 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 similarly to PFQ ensures prioritizing of streaming flows. The full description of PFQ and PDRR with the algorithms for packet queuing and scheduling are presented in [6] and [7].

The third FAN architecture, called Approximate Flow-Aware Networking is based on the AFD algorithm for packet scheduling. This solution assumes one FIFO queue for queuing packets which belong to elastic flows and prioritizing possibilities by using a separate FIFO queue for streaming flows. The AFD algorithm ensures fair transmission of elastic flows and lower complexity of XP routers in comparison to the other solutions.

The values of congestion control indicators in AFAN are estimated in a simpler way than in the other solutions. A counter incremented on the departure of each priority packet by its length in bytes is needed to compute the *priority_load*. Let $pb(t)$ be the value of this counter at time t , (t_1, t_2) is the measurement interval (in seconds) and C is the link bit rate. An estimate of the *priority_load* is:

$$priority\_load = \frac{(pb(t_2) - pb(t_1)) \times 8}{C(t_2 - t_1)} \tag{1}$$

The *fair_rate* is computed from the following formula:

$$fair\_rate = \frac{max\{S \times C, FB \times 8\}}{t_2 - t_1} \tag{2}$$

where *FB* is a number of bytes sent by elastic flows during the time interval $(t_1,\ t_2)$ divided by the number of elastic flows in PFL, *S* is the total length of inactivity in the transmission during the $(t_1,\ t_2)$ period, *C* is the link bit rate.

The queuing operation in AFAN is at the same complexity level as in FAN with PFQ or PDRR. However, packets are queued in a different way. Packets are stored in one of two FIFO queues (for elastic or streaming flows). A packet represents a streaming flow if it has less bytes than MTU in the queue. If a packet belongs to an elastic flow, the Approximate Buffer Size (*ABS*) parameter is computed form the following formula:

$$\begin{cases} ABS = (1 - w_q)ABS + w_q q & \text{if the queue is nonempty} \\ ABS = (1 - w_q)^m ABS & \text{if the queue is empty} \end{cases} \tag{3}$$

where w_q is queue weight, q represents the current buffer size and m is the number of packets that might have been transmitted by the router during the time that the line was free and is estimated from the following formula:

$$m = (time - q\_time)/s \tag{4}$$

where $time$ is the current time, q_time is the start time of the buffer idle time and s is the transmission time of a packet.

Two thresholds must be set in the FIFO buffer for elastic flows. If the ABS is greater or equal to the maximum threshold (max_th) the incoming packet must be dropped. If the ABS is greater or equal to the minimum threshold (min_th) and lower than the *max_th* the flow ID of randomly selected packet *d* from the elastic FIFO queue is compared with the flow ID of incoming packet p. If both packets represent the same flow, packet d is dropped and packet p is dropped with probability *P2* estimated in the same way as in the AFD algorithm. Packet p may also be queued if its flow ID is different than flow ID of packet d or ABS is lower than *min_th*.

Dequeue operation in AFAN is very simple. Firstly, the packets are selected from the streaming queue and then, if it is empty, from the elastic queue. This is a real gain in the algorithm complexity in comparison to PFQ or PDRR. In AFAN, there is no need to maintain the additional structure like Active Flow List (AFL) implemented in both known versions of FAN. We do not need to find a flow before sending a packet and update the content of AFL.

AFAN is scalable since the complexity of queuing algorithms does not increase with the link capacity. Compared to other QoS architectures, AFAN (and also both other versions of FAN) scalability, due to the lack of signaling and very low data handling complexity, is not matched by any other architecture [8]. Finally, AFAN 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 [9].

III. Intelligent routing algorithm for AFAN

In this section, a new algorithm for Approximate Flow-Aware Networks, which ensures intelligent routing possibilities in XP routers is proposed and analyzed. The pseudo-code of the algorithm is presented in Tab. I.

A new flow may be accepted at the admission control block in a congestion-less state. Then, based on the current routing table its identifier (ID) is added to PFL with the identifier of outgoing interface and the incoming packet is sent for queuing (lines 4-8). If the ID of flow represented by the arriving packet is in PFL, then the outgoing interface is found in PFL and packet is sent for queuing (lines 12-13).

In congestion, the situation is more complex. If the ID of flow represented by the incoming packet is in PFL, then operation taken in the XP router is similar to that described in the previous case. The outgoing interface is found in PFL and the packet is sent to the scheduler block (lines 18-21). On the other hand, a new temporal routing table must be determined. It has to be assumed that all congested links are treated as failed links at this time (line 25). The process of finding the temporal routing table may take some time. It depends on the routing protocol. For example, in OSPF it may take even tens of seconds. However, this process may be activated before the observed link becomes congested. The use of hysteresis may be a good solution to this problem. Moreover, the temporal routing tables may be written to the router memory and used in the future in a fast way. If the new routing table is determined, than based on it the ID of incoming flow is added to PFL with the ID of outgoing interface and the arriving packet is sent for queuing (lines 28-29). It is very important to note that after any topology change in a network (e.g., due to a link or node failure), the registrations in PFL have to be updated (lines 2-3 and 16-17). However, the mechanism which solves the problems when a failure occurs in a network is not considered in this paper.

The algorithm is based on the load balancing assumption. However, the additional operations are needed. The key point is that outgoing interfaces in the XP routers are selected based on registrations in PFL (in opposition to the current approach where outgoing interfaces are fixed based on the routing table). The new algorithm is a smart solution which does not increase operation complexity in the XP routers significantly and allows for improving transmission performance in Approximate Flow-Aware Networks.

TABLE I
PSEUDO CODE OF ADMISSION CONTROL WITH THE INTELLIGENT ROUTING ALGORITHM IN AFAN

```
1.  on a packet p of new flow F arrival in the congestion-less state
2.      If routing table has changed then
3.          update all IDs in PFL
4.      If ID(F) is not in PFL then
5.      begin
6.          add ID(F) to PFL
7.          based on routing table add ID(out int) to ID(F) in PFL
8.          send packet p for queuing
9.      end
10.     Else
11.     begin
12.         find in PFL the ID of outgoing interface for p
13.         send packet p for queuing
14.     end
**************************************************************
15. on a packet p of new flow F arrival in the congestion state
16.     If routing table has changed then
17.         update all IDs in PFL
18.     If ID(F) is in PFL then
19.     begin
20.         find in PFL the ID of outgoing interface for p
21.         send packet p for queuing
22.     end
23.     Else
24.     begin
25.         find new temporal routing table (assume that all
            congested links are failed)
26.         If new routing table has been found then
27.             begin
28.             based on new routing table add ID(out int) to ID(F) in PFL
29.             send packet p for queuing
30.             end
31.     end
```

A. Comparison of the intelligent routing for AFAN approach with existing solutions

The concept of an intelligent routing for Approximate Flow-Aware Networks is similar to load balancing. However, the proposed approach uses a new method for choosing the routes for packets which represent flows. Many reports in the literature have analyzed the similar problem. Three of them are briefly discussed and compared with the proposed solution in this section.

Load balancing is a popular method for improving the network performance. It is implemented in almost every routing protocol. When we have several paths to the destination node with equal cost, traffic is equally divided and sent through these paths. There are also several more advanced solutions implemented, e.g., in EIGRP, where load balancing is enable by using paths of unequal costs [10]. However, due to my best knowledge, there are no load balancing mechanisms which allow for dynamic change of paths' costs, e.g., based on traffic load in network links.

The Valiant's load balancing was introduced in [11]. It assumes that the outgoing interface may be randomly selected among all not congested ports. However, this solution works only in a fully-connected logical mesh network. The proposal presented in this paper does not assume such limitation and thus is more universal.

Traffic engineering in MPLS, deeply described in [12], assumes that paths for particular flows are determined using a link-state database which contains flooded topology and resource information. In this context, traffic engineering in MPLS is similar to the mechanism investigated in this paper. However, there is one important drawback of this solution which is eliminated in my proposal. In the intelligent routing for AFAN, the signaling protocol like RSVP or LDP and packets' labels are not needed what improves the network performance. Moreover, thanks to this advantage, the scalability of the mechanism proposed in this paper is also better.

In the following two paragraphs, the congestion control mechanisms which additionally improve the transmission performance of streaming flows in AFAN are presented.

IV. The RAMAF mechanism

The Remove and Accept Most Active Flows congestion control mechanism is the most recent proposal for FAN (and also for AFAN) to improve the performance of streaming flows [4]. The goal of it is to reduce the acceptance times of streaming flows in XP routers to the acceptable level. According to [13] the international streams (e.g., inter-continent voice calls) should begin transmission in 11 s, while the proper delay for local streams should not exceed 6 seconds. In basic AFAN, there are no acceptance guarantees for any flow. In the extreme case streaming flows may have to wait for transmission for a long time. The RAMAF mechanism assumes additional operations in the MBAC. They are illustrated in Fig. 2. In RAMAF, the IDs of a number of most active flows are periodically removed from PFL and written to Priority in Access Flow List (PAFL). This number is set dynamically based on the queue occupancy. Next, in a congestion-less state, the IDs from PAFL are added to PFL again. It allows for dynamic changes of values of the *fair_rate* parameter around the minimum acceptable value, and consequently gives chances to new flows to begin their transmissions.

V. The Simple Congestion Control Mechanism

In this paper, new congestion control mechanism for Flow-Aware Networks, called Simple Congestion Control Mechanism (SCCM) is introduced. The pseudo-code for implementation of this proposal is presented in Tab. II. The main

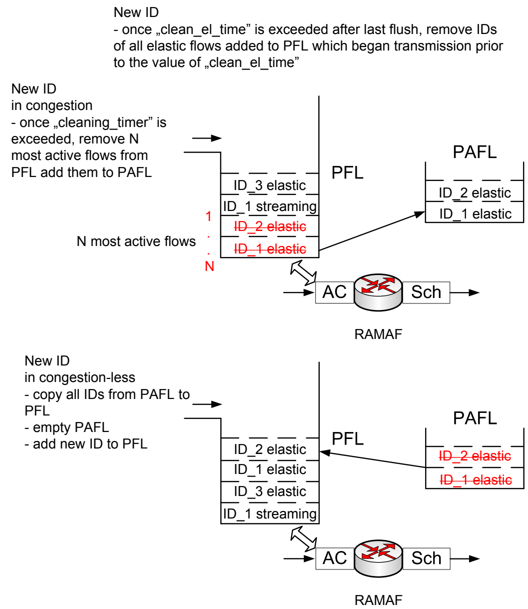

Fig. 2. Packet service in AFAN with the RAMAF mechanism

assumption of the mechanism is to eliminate congestion periodically for a short time and then to eliminate the IDs of elastic flows accepted during that short time. The goal is realized by periodical reducing the min_fair_rate value to 0 (line 24 in Tab. II). The time period is set by the $reduce\_timer$ value (line 22). After reducing the min_fair_rate value the new flows are accepted during the half of the measurement interval of the $fair\_rate$ value (lines 3-7). This time period is long enough to accept all waiting for transmission flows. After the time period given by the $clean\_el\_time$ parameter from the last reducing of the min_fair_rate, the identifiers of all elastic flows accepted during this interval are deleted from PFL (lines 8-20). The value of the $clean\_el\_time$ parameter should be estimated experimentally to eliminate all undesirable elastic flows. In the experiments provided in this paper, it was assumed that $clean\_el\_time$ was equal to three times of $fair\_rate$ measurement interval. Lines 30-32 ensure that the procedure works only in the congestion state.

VI. Simulation analysis

In this section, the results of carefully selected simulation experiments run in the ns-2 simulator are presented. The goal of them is to show how the new algorithm for intelligent routing in AFAN improves network performance. Moreover, the advantages and disadvantages of the RAMAF and the SCCM are presented by several simulation experiments.

120 simulation runs were provided for four network topologies. The most advanced topology is presented in Fig. 3 and hereafter will be called as topology no. 4. In topology no. 3 links L11-L15 are not present. Consequently, in topology no. 2 links L1-L6 are present and in topology no. 1 only one path between the source (S) and destination (D) nodes is implemented (through L1, L2 and L3 links). The goal of

TABLE II
PSEUDO CODE FOR REALIZING THE SCCM FUNCTIONALITY IN AFAN

```
on a new flow packet p arrival
    current_time = Scheduler :: instance().clock()
    If current_time − last_reduce > 0.5 ∗ fair_rate_int
    begin
        min_fair_rate = org_min_fair_rate;
        control_param2 = 0;
    end
    If current_time − last_reduce > clean_el_time
       && control_param1 = 1 then
    begin
        For (i = 1; i ≤ pfl_size; i++) do
        begin
            active_time(i) = current_time − first_time(i)
            If flow_bytes(i) ≥ MTU then
            begin
              If active_time(i) <= clean_el_time then
                remove ID(i) from PFL
                control_param1 = 0;
            end
        end
    end
*************************************************************
    *************** in congestion state ****************
    If current_time − last_reduce > reduce_timer then
    begin
        min_fair_rate = 0
        last_reduce = Scheduler :: instance().clock();
        control_param1 = 1;
        control_param2 = 1;
    end
    discard packet p
*************************************************************
    ************* in congestion-less state **************
     last_reduce = Scheduler :: instance().clock();
     proceed with packet p
*************************************************************
```

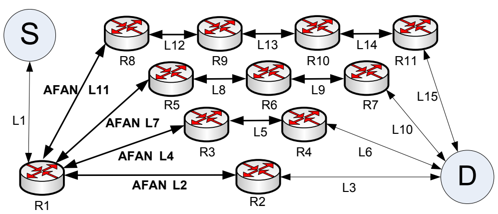

Fig. 3. Simulation topology

the experiments provided for each topology was to show how the volume of total traffic changes if the number of routes between nodes S and D increases after implementation of the intelligent routing algorithm in AFAN.

The simulated topologies are simple, yet adequate to analyze the new algorithms in AFAN. The reason behind such a statement is that all nodes in AFAN 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 algorithms.

It was assumed that the capacity of each AFAN link was set to 100 Mbit/s. Of course, this value is too low when considering core links, however the results obtained during the simulation analysis are scalable. The only reason to make simulations for low capacity core links is time needed for providing experiments. In our conditions one simulation run took about two hours. The capacity of other links was set to 1 Gbit/s. The simulations were repeated at least 10 times for each experiment. 95% confidence intervals were calculated by using the Student's t-distribution.

The traffic pattern with Pareto distribution for calculating the volume of traffic to be sent by each of 200 elastic flows from node S to D (the size of data to be sent was set to 150 Mbit and the shape parameter was set to 1.5) was provided. I used the exponential distribution for generating the time intervals between beginnings of the transmissions of elastic flows (the mean value of the inter-arrival time was set to 0.2 s). The exponential distribution was also used for generating the time intervals between beginnings of the transmissions of 20 streaming flows (the mean inter-arrival time was set to 1 s). 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 (like in a typical Skype VoIP connection). The elastic traffic was treated as the background traffic and used to saturate the analyzed links. One elastic flow which began its transmission when the simulation started and was sending its traffic during the whole simulation to the destination node D was also added. Packets of this flow were always sent through the longest route. For this flow, the goodput (mean rate of data successfully delivered to destination) was observed. The duration of each simulation run was set to 300 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 50 s.

For AFAN, min_th was set to 4000 packets and max_th to 9000 packets. For RAMAF, the $cleaning\_timer$ (minimum time between any flushing of PFL action) was set to 5 s. For SCCM, the $reduce\_timer$ was set to 5 s.

The results of the simulation experiments are presented in Tab. III and in Fig. 4 and Fig. 5. The simulation experiments provided in each topology show that the RAMAF and the SCCM allow for significant reduction of acceptance times of streaming flows. The obtained results are at the acceptable level (less than 6 s) in each case while in basic AFAN (without congestion control mechanisms) the streaming flows were accepted after dozens of seconds (see Tab. III). The results confirm that both the RAMAF and the SCCM are the good proposals for MBAC to reduce the acceptance times of streaming flows in each topology.

In Fig. 4, we can see that goodput in AFAN with RAMAF is worst than in basic AFAN. It is a result of periodical cleaning of the PFL content and in consequence breaks in transmission of elastic flows. Moreover, the value of goodput is almost the same in AFAN with RAMAF independently of topology. It is caused by the fact that the RAMAF cleans the content of the PFL of each link in the topology and, as a result, the observed flow has to slow down from time to time when the AFAN link on its route becomes congested. In basic AFAN the goodput increases with number of routes to the destination. The similar situation is observed for SCCM where IDs of elastic flows are not removed from the PFL. The observed flow slows down when congestion is periodically eliminated and new flows are accepted. However, its transmission is never broken and when we have more routes to the destination node the difference between the SCCM and basic AFAN is reduced. It confirms that the SCCM provides the better transmission properties for the elastic flows than the RAMAF. Moreover, it ensures similar acceptance times for the streaming flows.

The most interested results are presented in Fig. 5. We can see that with linear increasing of the number of routes to the destination node the amount of total transmitted traffic in a network also increases linearly. It is consistent with my expectations because the amount of data received by the sink should increase in proportion to the number of used links. This proves that the intelligent routing algorithm for the Approximate Flow-Aware Networks works as it was assumed. It allows for better usage of the available resources in a network and significantly improves performance in a network. This conclusion is true for basic AFAN as well as for AFAN with the RAMAF or the SCCM.

TABLE III
THE VALUES OF MEAN ACCEPTANCE TIME IN THE ANALYZED TOPOLOGIES

n **routes in topology**	**accept. time [s]** (Basic AFAN)	**accept. time [s]** (RAMAF)	**accept. time [s]** (SCCM)
1	114.22±25.02	0.63±0.63	1.70±0.22
2	82.37±18.00	0.91±0.73	1.03±0.63
3	85.81±11.90	0.47±0.47	0.65±0.49
4	77.00±11.78	0.31±0.29	0.94±0.66

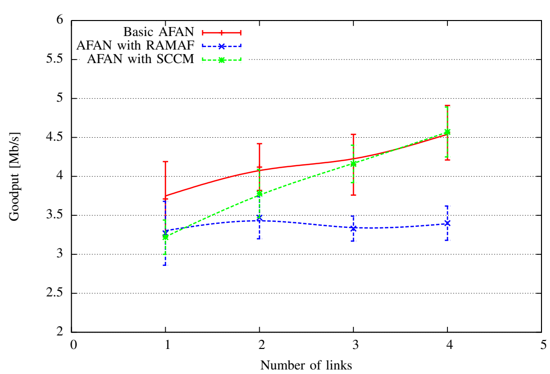

Fig. 4. TCP efficiency in the analyzed topologies

VII. Conclusion

Approximate Flow-Aware Networking is a newest promising concept for Flow-Aware Networks which may be used in the Future Internet. This simple solution is scalable and conforms to the net neutrality paradigms.

The SCCM and RAMAF congestion control mechanisms ensure fast acceptance times of streaming flows in XP routers. Moreover, the results of the simulation experiments prove that

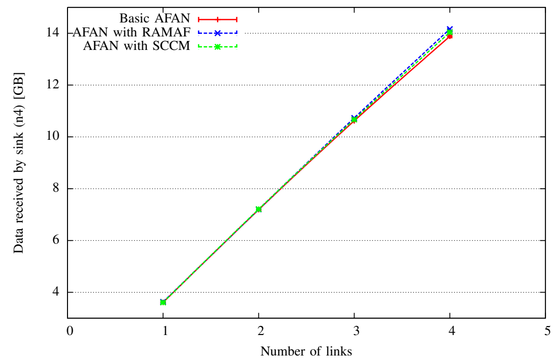

Fig. 5. TCP efficiency in the analyzed topologies

SCCM assures better transmission conditions for elastic flows than the RAMAF approach.

The new algorithm for intelligent routing in Approximate Flow-Aware Networks gives chances to improve transmission performance in a network significantly. It allows for using different routes to destination nodes. In this way the resources in a network may be utilized in a more efficient way.

The AFAN architecture with the SCCM and the intelligent routing algorithm is a complete solution to use in the scalable, reliable and efficient Future Internet.

Acknowledgements

The research was carried out with the support of the project "High quality, reliable transmission in multilayer optical networks based on the Flow-Aware Networking concept" founded by the Polish National Science Centre under the project no. DEC-2011/01/D/ST7/03131.

References

[1] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet Inter-Domain Traffic," in *ACM Sigcomm 2010*, New Delhi, India, August-September 2010.

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

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

[4] J. Domzal, N. Ansari, and A. Jajszczyk, "Congestion Control in Wireless Flow-Aware Networks," in *IEEE ICC 2011*, Kyoto, Japan, June 2011.

[5] 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.

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

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

[8] 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.

[9] J. Domzal, R. Wojcik, and A. Jajszczyk, "Qos-aware net neutrality," in *First International Conference on Evolving Internet, 2009. INTERNET '09.*, August 2009.

[10] "Load balancing data center services." Cisco Systems, 2004.

[11] R. Zhang-Shen and N. McKeown, "Designing a Predictable Internet Backbone Network," in *HotNets*, San Diego, CA, USA, November 2004.

[12] E. Osborne and A. Simha, *Traffic Engineering with MPLS.* Pearson Education, 2002.

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