Click-based tests of QoS mechanisms for flow-based router

1st Jerzy Domżał
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
jdomzal@kt.agh.edu.pl

2nd Edyta Biernacka
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
ebiernac@agh.edu.pl

3rd Jakub Dudek
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
jakub@dudek.in

4th Piotr Jurkiewicz
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
piotr.jerzy.jurkiewicz@gmail.com

5th Mirosław Kantor
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
kantor@kt.agh.edu.pl

6th Łukasz Romański
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
lukasz.romanski@wp.eu

7th Krzysztof Wajda
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
wajda@kt.agh.edu.pl

8th Robert Wójcik
Department of Telecommunications
AGH University of Science and Technology
Krakow, Poland
wojcik@kt.agh.edu.pl

***Abstract*—We demonstrate tests of a flow-based router enhanced with the additional mechanisms: Emergency Connections, Efficient Congestion Control, Global Protected Flow List and Per-User Fairness. The tests were conducted in the Click environment. The analyzed mechanisms allow for immediate acceptance of high priority traffic, protection of streaming flows in congestion, and ensure high reliability and fairness. In this paper, we show that these mechanisms work successfully in a laboratory network. Moreover, they all work at the same time providing high-level coordinated performance. Test results show significant advantages of the cross-protect router architecture over standard IP routers in several areas.**

***Index Terms*—Cross-Protect router; Flow-Aware Networking; Quality of Service; Click; scheduling**

I. Introduction

A Cross-Protect (XP) router is a device developed for Flow-Aware Networking (FAN), an Internet Protocol (IP) networks enhancement to efficiently provide quality of service guarantees. The main goal of FAN is to improve the current IP networks performance under heavy congestion. To achieve this goal, two traffic management mechanisms to control link sharing are proposed: measurement-based admission control and fair scheduling with priorities. The admission control mechanism is used to keep the flow rates sufficiently high to provide a minimal level of performance for each flow in case of overload. The scheduler realizes the fair sharing of link bandwidth while ensuring negligible packet latency for flows emitting at lower rates. Two types of flows are defined for FAN: streaming (priority traffic) and elastic. Packets of flows can be queued in the scheduler block according to one of three scheduling algorithms proposed for FAN. In Priority Fair Queuing (PFQ) [1], one Push-In First-Out (PIFO) queue is used and the packets of streaming flows are served first. In Priority Deficit Round Robin (PDRR) [2], packets of streaming flows are queued in one First-In First-Out (FIFO) queue while each elastic flow has its own FIFO queue. Queues for elastic flows are served according to the deficit round-robin regime. In Approximate FAN (AFAN), one FIFO queue is responsible for serving elastic flows and another FIFO queue serves priority flows [3]. The compendium of FAN was presented in book [4].

In [5], we have shown several advantages of FAN, proving that the main concept works. FAN can be considered as almost neutral architecture to IP. There are some proposals to extend its functionality or operation, e.g. in [6], the concept based on Software-Defined Networks (SDN) was proposed. However, there is stiil room for new mechanisms. In this paper, we

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9348111

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

discuss our experiences in building a Click-based prototype and implementing new mechanisms. The aim was to test the efficiency in real laboratory network conditions.

In the literature, several concepts were proposed to enhance FAN. However, all of them were validated through simulations only. The simulation analysis is usually provided with limitations and not all factors from real networks can be taken into account. For example, it is assumed that flow identifiers are written in a table, but it is not explained and analyzed how to do this in detail and whether it would be efficient. Moreover, it is usually hard to observe times of flows switching or implement real mechanisms to observe congestions in a simulator. Here, we analyze test results obtained during laboratory tests in which we used the prototype of an XP router.

The prototype was built in a Click modular router environment [7]. Fig. 1 shows a block diagram of Click in which several new blocks were added. These blocks (marked in color) implement FAN-related functions and new mechanisms which were tested. In comparison to the basic router configuration presented in [5], we added new flow lists, which allow us to implement new mechanisms (GPFL table, PUFL table). We also added the *CheckCongestionECCM* block, which is responsible for analyzing the congestion status of the link. All the lists in the router are analyzed depending on the status of the outgoing link.

We selected four mechanisms to evaluate: Emergency Connections, Efficient Congestion Control, Global Protected Flow List and Per-User Fairness. These mechanisms were chosen because they assure fair and reliable transmission and priority treatment for streaming traffic.

II. Related Work

FAN is a networking architecture which was proposed to send traffic with assured quality of service in a simple manner in a distributed way. This proposal can be considered as a competitive architecture to the Software-Defined Networks (SDN). While SDN have benefits in network management as a result of the central controller implementation, the FAN minimizes the signalling traffic and is more reliable due to the lack of the controller being a single point of failure. The networking architectures managed in a distributed way and based on flows had been proposed in the past, however they were implemented only partially or even stayed only as proposals. For example, Caspian Networks tried to add flow-based routing to basic IP routers in the years 2000-2006, but this concept failed. However, since that time devices became more efficient and it has been proved that flow-based solutions can be implemented even in a wide networks, e.g. in B4 Google network [8].

The Flow-Aware Networking concept has attracted worldwide attention in recent years, which has resulted in many studies. Several mechanisms have been proposed to improve FAN functionality and operation. These were related to observed technical problems with the implementation or performance. For example, the authors of [9] proposed a flow-aware approach to evaluate probability of compromise in

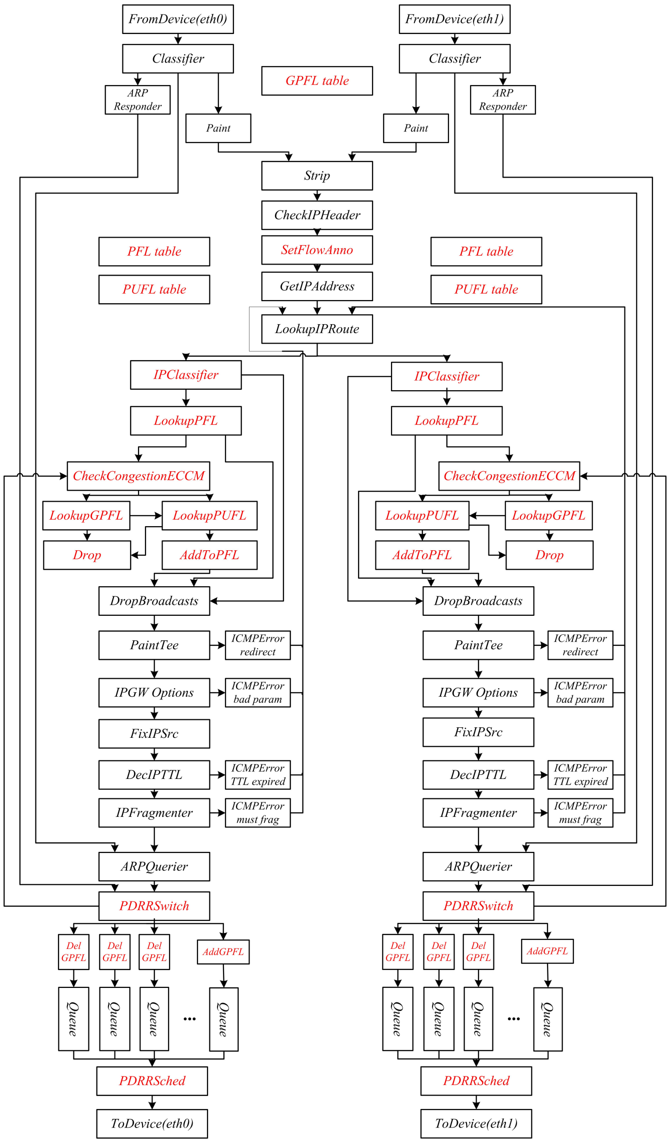

Fig. 1. XP router with advanced mechanisms - block scheme.

combine structure network. The flow-aware congestion control mechanism for datacenter networks was presented in [10]. The authors of this paper proposed algorithms to select flows based on assumed criteria and to provide different congestion control using explicit congestion notification (ECN). FAN concept was also analyzed as a possible architecture for cellular networks. The performance analysis of FAN in LTE backbone was presented in [11]. The compendium of FAN was presented in book [4].

When the link is congested, the admission control block in the XP routers block new flows which increases their acceptance delay. Two approaches solving this problem have been proposed. The first one is the Static Router Configuration which helps with the transmission of emergency calls [12]. The second approach assumes periodic partial or total clearing of the PFL content. Many modifications of this method were analyzed and presented in [13] or [14].

The Multilayer Flow-Aware Networking (MFAN) was introduced in [15]. It is shown that FAN can operate in multilayer environment enhancing the quality of service for flows in this way. In this proposal, additional optical resources are available for the routers. They can be set up when congestion is observed at the IP layer. Three admission control policies deciding which flows should be served at the IP layer and which ought to be redirected to the optical one, i.e., Newest Flow Policy, Oldest Flow Policy and Most-Active Flow Policy have been proposed. In [15], the authors explain these policies in detail and compare their performance by evaluation.

FAN can be considered as almost neutral architecture to IP. It does not interfere with IP protocol functionality, including routing procedures. However, there is a proposal to introduce a new routing scheme. In [16], the intelligent routing has been proposed and implemented. It clearly improves network performance especially in overload and failure conditions.

III. Emergency Connections Mechanism

Although providing superior transmission quality, FAN may force flows to wait for the network resources. Such a situation is inconvenient for the realization of emergency VoIP connections. To overcome the presented problem, an Emergency Connections Mechanism (ECM) based on differentiated (selective) blocking algorithm can be implemented in FAN. It is simple, adequate, and above all, a feasible solution to classify, select and serve critical connections.

In ECM, classifying flows as emergency is based on source and destination IP address. IPClassifier has been placed right before the LookupPFL (see Fig. 1) element and configured in a way that packets with an emergency source IP address are passed for further processing without any admission control procedures, while any other packets are sent to the admission control block for PFL lookup. Quick and easy ECM implementation is a result of Click's flexibility.

IV. Efficient Congestion Control Mechanism

The main aim of the Efficient Congestion Control Mechanism (ECCM) mechanism is to minimize acceptance delays of streaming flows, which usually are real-time transmissions such as voice or video connections. According to [17], local voice connections should be accepted in 6 s while international ones in 11 s. In standard FAN, when the outgoing link is congested, they have to wait even over hundreds of seconds before they start their transmission.

The ECCM mechanism starts when the outgoing link is congested for at least max_accept_delay time and a new flow wants to begin its transmission. In such a case FR (current fair rate estimated by the router) is set to $minFR$ (minimum threshold of FR). Moreover, new values of FR are not estimated for a period of $0.5 \times FR\_interval$ (half of time interval between two estimations of the fair rate values). It ensures that the link becomes uncongested (when FR is lower than $minFR$ the link is considered as congested). $0.5 \times FR\_interval$ is sufficient to accept all flows waiting begin transmission. The value of the max_accept_delay parameter should be set statically, e.g., to 6 s. Finally, after time equal to the interval between two estimations of the fair rate values from the starting point of the ECCM, it is necessary to remove, from PFL, all identifiers of elastic flows added when fair rate was equal to min_FR.

For the purpose of ECCM implementation, previously developed *CheckCongestion* element has been replaced witha new *CheckCongestionECCM* element implementing all the ECCM rules with configurable max_accept_delay parameter (see Fig. 1). A new ECCM table has been created to store all the flows admitted by ECCM during congestion. This way, we are able to identify which elastic flows admitted by ECCM should be removed from PFL when required.

V. Global Protected Flow List

In a standard XP router, for each outgoing link a separate PFL is implemented. The concept of Global Protected Flow List (GPFL) assumes that one additional list is implemented in each router with at least two outgoing links. This list contains identifiers of all flows active in all outgoing links, while streaming flows are marked with high priority. When a link fails, all streaming flows from this link can continue transmission in the backup link without breaks (if only the outgoing link is not saturated). When a routing protocol calculates a new path, identifiers of streaming flows are accepted immediately on this path.

A few simplifications, as compared to the original GPFL concept, were introduced to optimize its implementation. We have dropped the concept of putting both elastic and streaming flows to GPFL with different priorities. In our implementation only streaming flows can be added to GPFL as these are the only type of flows that can benefit from it. Such an approach implies that the type of flow has to be known when the decision about adding to GPFL is being made. Therefore, elements *AddGPFL* and *DelGPFL* are responsible for adding and deleting flows from GPFL respectively. They have been moved to the scheduling block of XP router. *AddGPFL* is connected to the input of the priority queue while *DelGPFL* is connected to the input of standard queues for elastic flows. As a result, when an initially streaming flow becomes elastic over time, it is no longer kept on GPFL.

Besides these two elements, the *GlobalFlowTable* element containing the table itself has been added. Most of the implementation has been reused from an already developed PFL table. The last element required for the GPFL mechanism is *LookupGPFL* which verifies the presence of a flow on the list. It is used when the flow is not on a PFL and congestion is detected. If the flow is not on GPFL as well, it is discarded. Otherwise, it is admitted for further processing. The location of the new GPFL elements in the router block scheme is shown in Fig. 1.

The pseudo-code for realizing the GPFL concept in FAN is provided in Fig. 2.

```
on a packet $p$ of flow $F$ arrival
 if ID($F$) is in the PFL then
begin
    if $F$ is streaming then
        add ID($F$) to the GPFL
        send $p$ for queuing
end
 else (not in the PFL)
begin
      If link is congested then
      begin
          if ID($F$) is in GPFL then
          begin
              add ID($F$) to PFL
              send $p$ for queuing
          end
          else drop $p$
      end
      else (link not congested)
      begin
              add ID($F$) to PFL
              send $p$ for queuing
      end
end
```

Fig. 2. Pseudo-code for realizing the GPFL concept in FAN

VI. Per-User Fairness

The per-user fairness (PUF) works in a relatively simple way. When a packet of the new flow arrives at the router, it is checked how many flows have been accepted in current measurement interval of FR. It is assumed that maximum N flows can be accepted for a user. Therefore, the incoming packet is dropped if the number of accepted flows for its source is higher than N. In the other case, the counter of admitted flows is increased, an identifier of flow related to the incoming flow is added to PFL and the packet is sent to the correct queue in the scheduler. Each time a new value of FR is estimated, counters for all sources are set to zero. The value of N is assumed by the operator. This value should be estimated based, e.g. on simulation experiments and should ensure fair access to the resources.

The following two new elements have been added to implement PUF in Click: the PUFL table and *LookupPUFL* (see Fig. 1). The PUFL table implements the table itself. The *HashTable* data type available already in Click has been used to build the table. It is indexed by source IP addresses and stores the number of active flows for each source. The *LookupPUFL* element serves packets of flows not present on PFL right before the addition to PFL. It has configurable parameter N (which means the maximum number of flows allowed for a user). It discards a new flow when the number of flows for a particular source exceeds N. Otherwise, a flow is added to PFL and further processed. In addition to these two elements, a new function has been added to the *PDRRSched* element responsible for cleaning the PUFL table at the end of the fair rate measurement interval (garbage collection mechanism). The implementation of this mechanism needs to add one table. The content of this table is analyzed for each new flow. As a result, the resource usage is negligible.

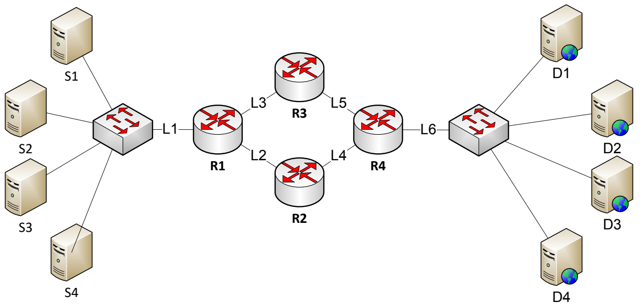

Fig. 3. Testing topology.

VII. Tests of the XP Router with Advanced Mechanisms

The results of tests conducted in the laboratory are presented in this section. The main aim was to present advantages of mechanisms described in previous sections implemented in the XP router.

A. Testing methodology

We tested our prototype router in a laboratory network with real traffic generated in network nodes. The basic network topology used in our tests is presented in Fig. 3.

For each implemented mechanism, some basic functional tests as well as more advanced numerical evaluation were performed. Finally, all mechanisms were connected together in a single configuration file and tested.

Traffic was generated in source nodes S1-S4. These were PC computers generating traffic. We used the D-ITG generator to generate traffic and to collect statistical data. The capacity of all core links was 10 Mbit/s, while the capacity of access links was 100 Mbit/s. Each testing experiment was repeated several times (at least five times) which allowed us to collect statistically credible results.

In tests conducted by us, the used XP router prototypes were implemented on relatively slow PCs (with a single-core processor platform and with two external network cards). Therefore, we were not able to analyze the mechanisms' scalability. In a router with a 1 Gbit/s network card we were able to serve traffic with the maximum speed of 300 Mbit/s.

B. Tests in laboratory

To validate the ECM mechanism, the outgoing link L2 of R1 was put into permanent congestion state and then two hosts (S1 and S2) transmitted streaming flows to the destination nodes D1 and D2, respectively. The one with emergency source IP address (originated from S1) was admitted despite congestion while the second one (originated from S2) was rejected according to expectations.

The main emergency connection mechanism evaluation consisted of measuring acceptance delays of flows in the network. The background traffic consisting of a different number of concurrent elastic flows was generated in all sources and was sent to all destination nodes. Background flows sizes were generated from the Pareto distribution with 2 MB as the mean

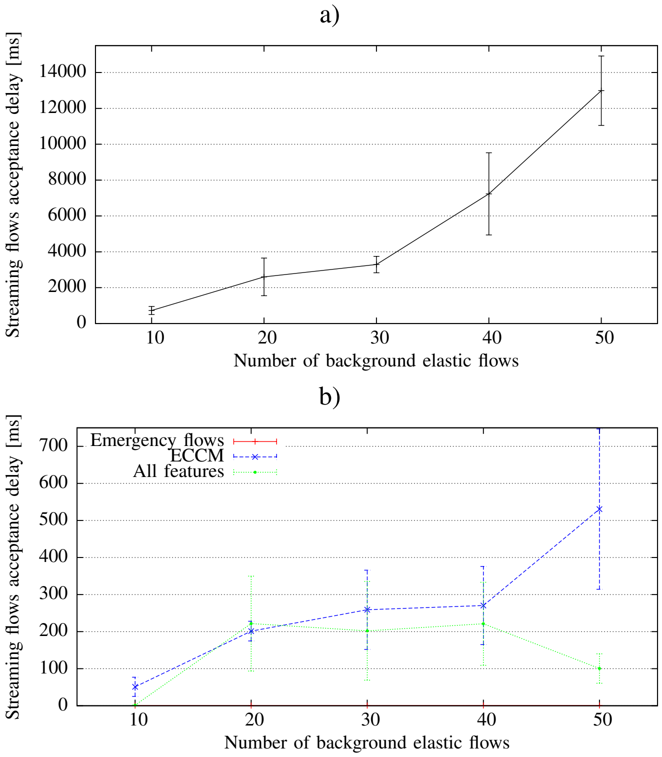

Fig. 4. Streaming flows acceptance delay for different number of background elastic flows: a) standard XP router b) standard XP router with advanced mechanisms.

value to send and shape factor equal to 1.5. Intervals between flows were generated from an exponential distribution but the mean interval value varied between the scenarios which resulted in different number of elastic background flows. Tests were performed in five scenarios with different number of elastic flows as the background traffic. 95% confidence intervals were calculated using the Student's t-distribution. The results of this test are shown in Fig. 4b. Similar results but for the standard non-emergency streaming flows may be observed in Fig. 4a.

The obtained results agree with the emergency connection mechanism assumption. Acceptance delays of emergency flows are lower than 1 millisecond, i.e., they are negligible irrespectively of the number of elastic flows transmitted in the background. We, therefore, conclude that emergency connection feature fulfills its role and works fully as intended accepting emergency flows immediately. To compare, the non-emergency streaming flows acceptance delay is much higher and increases with the number of background elastic flows reaching almost 13 seconds for 50 background flows (see Fig. 4a).

The tests of the ECCM mechanism were performed to observe acceptance delays of streaming flows as well. Traffic was generated as in the previous case. After introducing congestion and keeping it for max_accept_delay, we have verified that both streaming and elastic flows are temporarily accepted but elastic flows are rejected after fair rate calculation interval. Acceptance delay was measured for scenarios of different number of elastic flows in the background. Results are presented in the same Fig. 4b for comparison. The max_accept_delay parameter was set to 2 seconds and, as expected, the acceptance delay never exceeded it in case of the ECCM mechanism.

When compared with Fig. 4a and results for the standard XP router, significant advantage of the ECCM mechanism introduction is visible as far as streaming flows acceptance delay is considered. The acceptance delays for streaming flows increase with the increasing number of elastic flows in the background, however, they are still significantly lower than those for the basic XP router in the observed range. Moreover, transmission of elastic flows active in the background was not affected (more packet drops or higher delays were not observed).

GPFL functional test scenario was to execute failure of primary outgoing interface for a set of flows while the secondary interface is congested. As expected, streaming flows have been accepted without significant delay on a new path, while elastic ones had to wait for congestion to end. For numerical evaluation, comparison of the flow switchover times from the primary (R1-R2-R4) to the secondary (R1-R3-R4) path for the standard XP router and XP router with GPFL has been done. Secondary path was loaded with traffic that was, periodically, introducing congestion. The switchover time for standard XP router is the same for elastic and streaming flows and strictly depends on whether switchover occurs during congestion (flows have to wait until it ends) or not (flows are admitted immediately).

On average, the switchover time was equal to 2105.96±209.45 ms. For a router with GPFL implemented, average switchover time for elastic flows was very similar and equal to 2176.67±6.97 ms. However, as expected, a significant difference was observed for streaming flows, for which the switchover time was very low for a router with GPFL, namely 3.89±0.35 ms. Introduction of GPFL significantly improves FAN network reliability for streaming flows.

Per-user fairness functional tests were performed in scenario with sources which generated different number and types of flows and had N set to a fixed value. When the number of generated flows exceeded this value within FR measurement interval, new flows from a particular source were rejected by the router according to the expectations.

As far as numerical assessment is considered, the per-user fairness mechanism was tested in the environment which consisted of three sources (users) generating flows with different mean intervals (exponential distribution was used). User 1 (S1) was transmitting with the highest interval while User 3 (S3) with the lowest. Tests were performed in two scenarios: for an XP router with and without the per-user fairness feature. In case of the XP router with per-user fairness, router R1 was configured to accept maximum 2 flows per user in one fair rate computing cycle (4 seconds was set for this test). The measurement time was 40 seconds in both cases. Fig. 5 shows

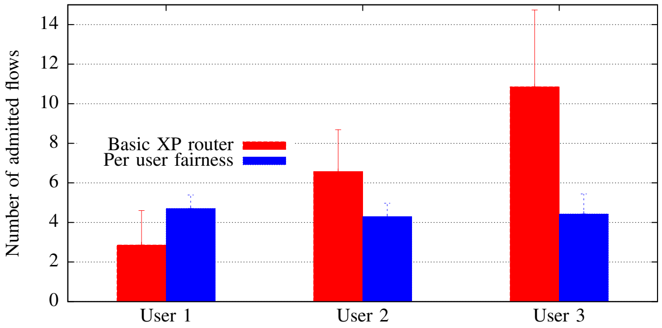

Fig. 5. Number of admitted flows within measurement interval for users transmitting with different frequencies.

a comparison of the number of accepted flows for each host in both scenarios during the measurement period.

The XP router without the per-user mechanism accepts flows according to FAN concept, so the largest number of flows has been accepted from the host which had the shortest mean interval. The router with the PUF mechanism worked fully fair and accepted similar number of flows for each user. It is worth to note that the total number of accepted flows is higher in the basic XP router scenario as PUF rejects some flows from aggressively transmitting sources besides standard congestion based rejection as in the XP router.

Finally, all implemented mechanisms have been tested together in a single configuration file. As part of functional testing we have repeated validation for each of the mechanisms to show that it is possible to implement them in one network. As part of more advanced testing, streaming flows acceptance delay has been measured in the same scenarios as for ECCM, PUF and the basic XP router. As can be observed in Fig. 4b, results are even better than for ECCM alone. This is due to the fact that when PUF and ECCM are working together, PUF by rejecting too extensive traffic from a particular source creates a better environment for streaming flows acceptance improved by ECCM, ensuring at the same time acceptable transmission performance for accepted elastic flows.

VIII. Conclusion

We showed that the proposed respective mechanisms to enhance FAN operation implemented alone as well as in a coordinated approach. These mechanisms all working at the same time provide the improved performance. Test results are very promising and show significant advantages of the XP architecture over standard IP routers in several areas.

Conducted work shows the advantages of the FAN concept. Continuous development of an XP router enables further FAN tests and experiments to be performed in the laboratory or even in real networks. We believe that tests presented in this paper will accelerate the research on Flow-Aware Networks and will open the window to conduct research also on partially centrally-managed flow-based networks.

Acknowledgement

The research was carried out with the support of the project ”Intelligent management of traffic in multi-layer Software-Defined Networks” founded by the Polish National Science Centre under project no. 2017/25/B/ST6/02186

References

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

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

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

[4] J. Domżał, R. Wójcik, and A. Jajszczyk, *Guide to Flow-Aware Networking: Quality-of-Service Architectures and Techniques for Traffic Management*, Computer Communications and Networks; ISSN 1617-7975, Springer, 2015.

[5] J. Domżał, J. Dudek, P. Jurkiewicz, L. Romański, and R. Wójcik, “The cross-protect router: implementation tests and opportunities,” *IEEE Communications Magazine*, vol. 52, no. 9, pp. 115–123, September 2014.

[6] J. Domżał, R. Wójcik, and E. Biernacka, “Efficient and reliable transmission in Flow-Aware Networks - an integrated approach based on SDN concept,” in *ICNC 2017*, Jan 2017, pp. 837–842.

[7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The Click modular router,” *ACM Trans. on Computer Systems*, vol. 18, pp. 263–297, August 2000.

[8] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, “B4: Experience with a Globally-deployed Software Defined Wan,” *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug. 2013.

[9] O. Yeremenko, M. Yevdokymenko, and A. Persikov, “Flow-aware approach of evaluating probability of compromise in combined structure network,” in *2017 2nd International Conference on Advanced Information and Communication Technologies (AICT)*, July 2017, pp. 258–261.

[10] C. Lee, Y. Nakagawa, K. Hyoudou, S. Kobayashi, O. Shiraki, and T. Shimizu, “Flow-aware congestion control to improve throughput under tcp incast in datacenter networks,” in *2015 IEEE 39th Annual Computer Software and Applications Conference*, vol. 3, July 2015, pp. 155–162.

[11] A. Sniady and J. Soler, “Performance of Flow-Aware Networking in LTE backbone,” in *Proceedings of OPNETWORK 2012. OPNET*, 2012.

[12] A. Jajszczyk and R. Wojcik, “Emergency calls in flow-aware networks,” *Communications Letters, IEEE*, vol. 11, no. 9, pp. 753–755, September 2007.

[13] J. Domżał and A. Jajszczyk, “New Congestion Control Mechanisms for Flow-Aware Networks,” in *IEEE ICC 2008*, Beijing, China, May 2008.

[14] J. Domżał, 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).

[15] V. Lopez, C. Cardenas, J. A. Hernandez, J. Aracil, and M. Gagnaire, “Extension of the Flow-Aware Networking (FAN) architecture to the IP over WDM environment,” in *Proceedings of 4TH Int. Tel. Net. Workshop on QoS in Multiservice IP Networks 2008*, Venice, Italy, February 2008.

[16] J. Domżał, “Intelligent routing in congested approximate flow-aware networks,” in *IEEE Global Communications Conference (GLOBECOM), 2012*, Dec 2012, pp. 1751–1756.

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