The Cross-Protect Router: Implementation Tests and Opportunities

Jerzy Domżał, Jakub Dudek, Piotr Jurkiewicz, Łukasz Romański, and Robert Wójcik

The authors are with AGH University of Science and Technology.

ABSTRACT

The concept of flow-aware networking (FAN) has been introduced, discussed, and analyzed only with the use of mathematical models or computer simulations. This article presents the results of FAN's first practical implementation. For the environment, the Click Modular Router was chosen. In the article, vital implementation-related issues are shown and analyzed. The results of post-implementation tests are also presented. They confirm the advantages of the XP router over the IP router. We prove that traffic in FAN is served fairly, and the packets of streaming flows are transmitted with high priority. Moreover, the test results confirm that streaming flows are served with acceptable quality even in highly loaded links, which is not observed in the classic IP network. The tests also show several problems not caught by the simulative studies available in the literature. First of all, to increase efficiency, a flow list had to be implemented in a different way than originally proposed. Also, a queuing algorithm had to be altered to eliminate too frequent changes of flow states, which resulted in a lack of stability. The presented tests ultimately prove that FAN works as a concept and that, by reaching maturity, it is ready for large-scale deployment.

INTRODUCTION

The concept of flow-aware networking (FAN), introduced in [1] as a quality of service (QoS) architecture, has attracted a lot of attention over the years and is still an interesting issue for researchers [2–4]. The idea was first presented as a notion and a set of guidelines, followed by mathematical analysis and extensive simulation studies. In due course, many additional mechanisms were proposed that enhanced the basic concept. However, until now, no one has built the cross-protect (XP) router — a router for FAN networks.

In this article, we show our experience of building the XP router. We show that additional nonstandard functions of XP routers are not complicated to implement, and can be implemented without excessive additional resources. Moreover, based on post-implementation tests, we show real pros and cons of FAN. The conducted experiments prove the usefulness and high efficiency of FAN.

The rest of the article is organized as follows. First, we introduce the reader to the basics of the FAN concept, with special attention devoted to issues discussed later in the article. We then present the Click Modular Router environment, which was chosen as our implementation framework, and explain why Click met our requirements. Next, we describe the steps undertaken during the FAN implementation process in Click. We provide the results of post-implementation tests — the first ever tests of FAN. In this section, a comparison between FAN and standard IP networks is also presented, and the efficiency of FAN is analyzed. Finally, we conclude the article.

FLOW-AWARE NETWORKING

FAN is a noteworthy QoS architecture that appeared as an alternative to the well-known *integrated services* and *differentiated services*. Its main characteristics include simplicity, scalability, and the ability to effectively deliver differentiated quality even in the presence of congestion. The main notion of FAN is to implicitly assign packets to more general flows; that is, groups of packets that have the same proper protocol field that consists of two IP addresses, two TCP or UDP port numbers, and the transport protocol identification. Because of the recognition of flows, traffic engineering is performed on the whole flow rather than on single packets. Treating traffic with the recognition of flows is a popular notion. Many current and past architectures follow this method [5].

There are only two types of flows in FAN: streaming and elastic. This classification is implicit, which means it is based only on the current flow rate. All flows emitting at lower rates than the current fair rate (explained later) are referred to as *streaming flows*, and packets of those flows are prioritized. The remaining flows are referred to as *elastic flows*. Streaming flows have priority over elastic ones because of the assumption that they are more vulnerable to sudden transmission rate drops.

To implement FAN in an IP network, the XP router is introduced. An XP router adds only two blocks to the standard IP router: measurement-based admission control and fair queuing

0163-6804/14/$25.00 © 2014 IEEE

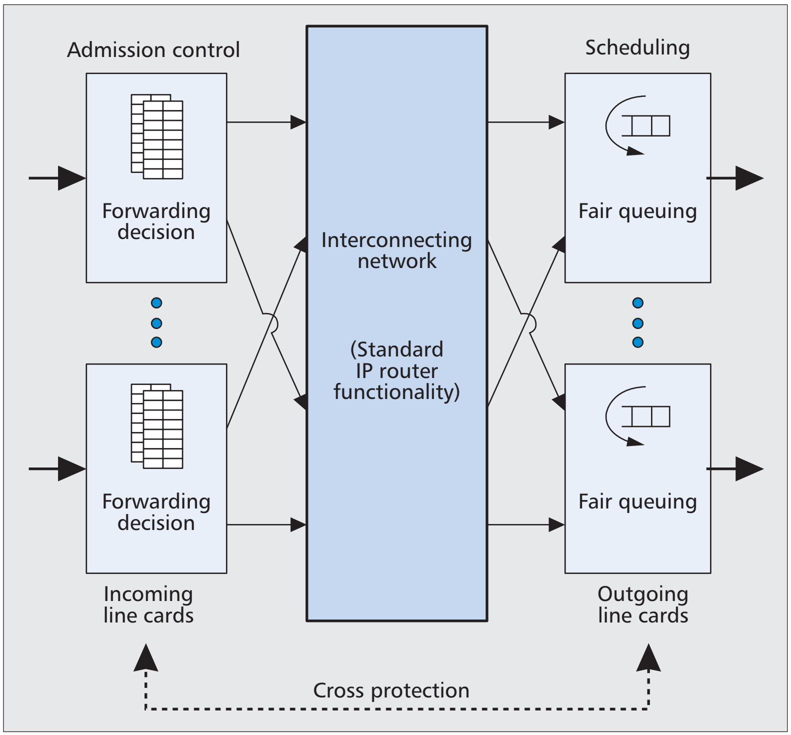

Figure 1. Cross-protect router diagram.

with priority blocks. These two blocks cooperate to provide maximum efficiency and scalability. Figure 1 presents a diagram of an XP router. Admission control elements are responsible for maintaining the protected flow list (PFL) — a list of currently active flows, the packets of which are forwarded regardless of current link congestion status. The packets of new flows (not present on the PFL) are forwarded only when the outgoing link is not congested. In this way, flows which have already been admitted preserve uninterrupted service.

The scheduling block is responsible for applying service differentiation, assuring fairness, and feeding link congestion measurements to the admission control block. It uses one of three queuing algorithms suitable for FAN: Priority Fair Queuing (PFQ), Priority Deficit Round Robin (PDRR), and Approximate Fair Dropping (AFD). All of these are designed to provide preferential treatment to streaming flows while dividing the residual capacity equally among elastic flows. The queuing algorithms have their own way of performing congestion measurement. In either case, two indicators are measured periodically: fair rate (FR) and priority load (PL) [6]. Following [7], fair rate "represents the amount of link's bandwidth, which is guaranteed to be available for a single flow, should it be necessary," and the priority load can be understood as "the amount of data that is prioritized."

Designating FAN devices as XP routers is a result of the mutual cooperation and protection that exists between both extra blocks. The admission control block limits the number of active flows in a router, which essentially improves the queuing algorithm's functionality and reduces its performance requirements. It is vital that queuing mechanisms operate quickly, because for extremely high-speed links the available processing time is strictly limited. On the other hand, the scheduling block provides admission control with information on congestion status on the outgoing interfaces. This mutual protection contributes to a smaller PFL, which significantly improves FANs' scalability.

Over the years, as the concept of FAN attracted attention, many studies that enhance the architecture or propose new mechanisms have appeared. For example, the problem of overly long waiting times in wireless environments was resolved in [8]. In [9], the difference between per-user and per-flow fairness is exposed. The flushing mechanism is presented in [10], and various possible admission control policies are shown in [11]. A method of route caching by using the PFL is presented in [12]. FAN is also a QoS architecture that fits within the network neutrality boundaries, as shown in [13].

Click Modular Router

Click Modular Router (or Click for short) is a suite for building flexible and configurable software packet processors. It is a Linux-based environment originally developed at the Massachusetts Institute of Technology (MIT) with subsequent development at Mazu Networks, the Institute for Communications and Information Research (ICIR), the Univeresity of California at Los Angeles (UCLA), and Meraki. It is widely used for building experimental software routers and switches. Click is a perfect platform for researchers to experiment with novel protocols and, as such, was chosen as an implementation environment for an XP router. Among its many advantages, those that determined its selection were great flexibility, ease of adding new features, clear and scalable architecture, and relatively high performance.

Flexibility in Click is achieved due to its modularity. Click is designed in an object-oriented way, and assembled from fine-grained packet processing modules called *elements*, which are technically C++ classes. Each individual element performs a simple operation on a packet, like queuing or decrementing a packet's time to live (TTL) field. Each element has input and output ports, which serve as the endpoints of connections between them.

A user builds a router configuration from a combination of elements by connecting them into a directed graph. Packets flow between elements, along the graph's edges. The behavior of a router is determined by choosing the elements and connections among them. A variety of elements is delivered by default with the Click distribution. The user can simply compose them in many different ways, but can also create new ones tailored to his needs. In addition, creating new elements is relatively quick and easy.

Click may run in two different modes: kernel mode and user-level mode. In kernel mode, Click runs as a module in the Linux kernel. The Click module replaces the operating system (OS) networking stack, so the OS does not handle the

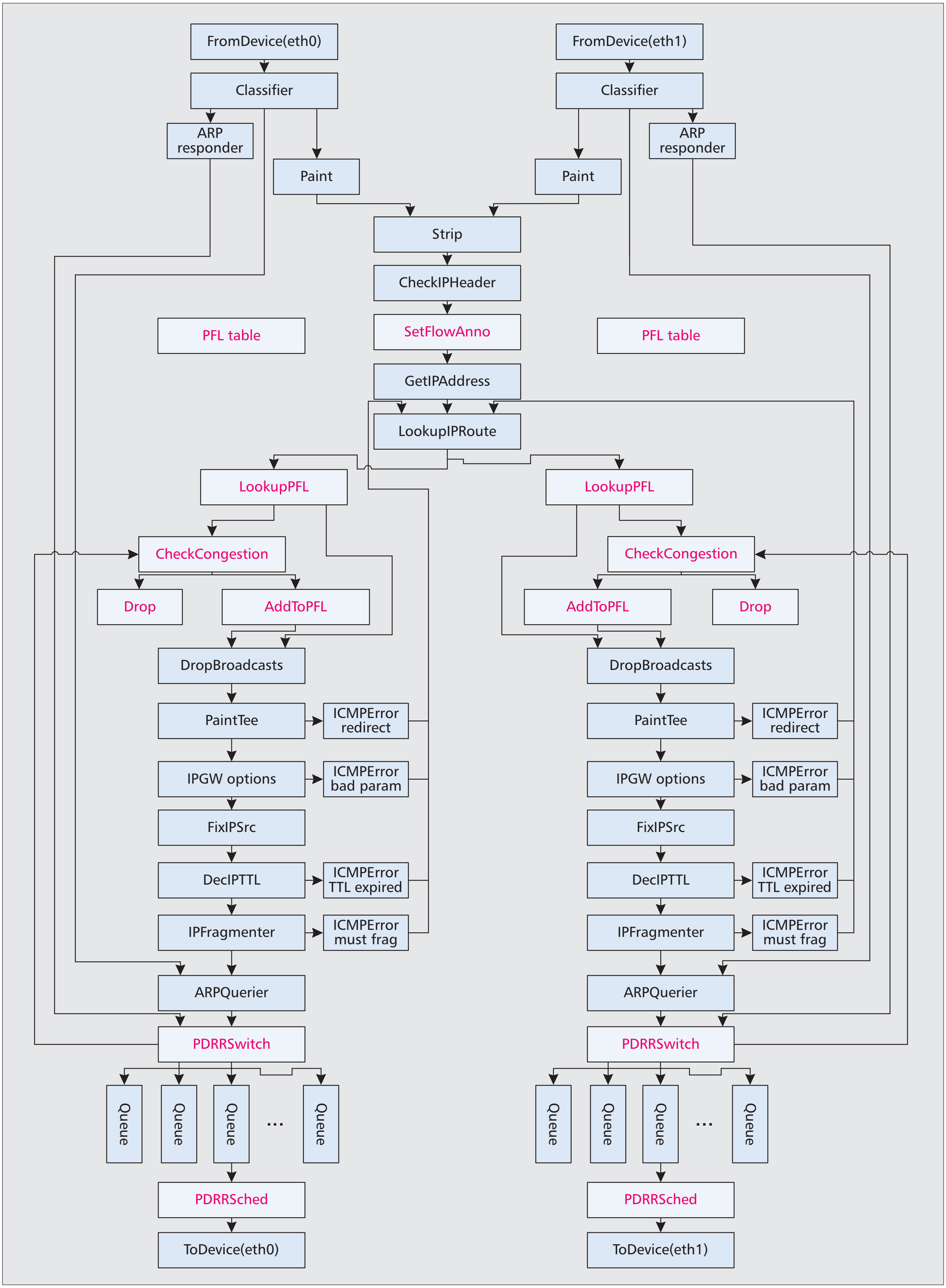

Figure 2. Router elements in a Click configuration block diagram (marked blocks are FAN related).

packets. Packet processing is done solely by the Click code running inside the kernel. Therefore, this mode is used when greater performance is required.

The implementation of the XP router in the Click environment required the development of several new Click elements and the preparation of a new router configuration file that combined these elements with the originally available IP router configuration.

It is worth mentioning that Click does not implement dynamic routing protocols but only simple static routing. However, dynamic routing can be used in Click. One way is to integrate Click with an external routing daemon such as XORP, an open source IP routing software suite that can fill up Click's routing tables automatically. Another possibility is to redirect packets to the OS to perform routing decisions.

A configuration that provides basic IP router functionality has already been developed by the Click authors, and is included in the Click environment. This configuration was used as a starting point for implementing FAN functionality. The implementation steps are described in the following section.

Implementation of the XP Router

The implementation of the XP router in the Click environment required the development of several new Click elements, and the preparation of a new router configuration file that combined these elements with the originally available IP router configuration. The new elements realize various functions. The elements added to the standard router configuration file are presented in grey boxes with red names in Fig. 2. First, in order to effectively identify different flows in a router, an element calculating the flow identifier was created (SetFlowAnno). We decided to use a hash function value for input arguments such as source and destination IP addresses, transport layer protocol type, and source and destination port numbers for each packet. The identifiers calculated this way are used to index flows on the PFL. PFL is the main component of the admission control block, which has been developed as a separate element in Click. A further two new elements are responsible for operations on the PFL: one for searching of the PFL content and verifying the presence of the proper flow (LookupPFL), and one for adding new identifiers of flows to it (AddToPFL). However, adding a new flow identifier to the PFL is possible only when the outgoing link is not congested. The last admission control element implemented by us in the Click environment is responsible for verifying the status of the outgoing link, based on information received periodically from the scheduling block (CheckCongestion).

During the implementation of the admission control block, some practical problems were encountered. One of the most important concerns was to decide on the proper PFL implementation type. One approach is to use an associative array, which contains the identifiers of flows in progress and stores full information about them. In this solution, an identifier is calculated as a simple concatenation of the header fields mentioned before. It ensures full flow differentiation and enables the possibility of implementation of some additional PFL applications, such as route caching (routing information may be stored in the PFL; hence, a routing decision has to be made only once for the first packet of the flow). The associative array itself can be implemented as a hash table with a collision resolution (e.g., a chained hash table) or a binary search tree. However, in this approach, it has to be ensured that PFL is regularly cleaned from expired flows entries (some kind of garbage collection needs to be implemented). The second, much simpler, approach assumes the use of a directly addressed flow list. This is a one-dimensional table of predefined size, indexed with flow identifiers, in which the timestamps of the last packets related to the flows are stored. Verification of a flow presence on the PFL requires only comparison of the proper timestamp with the current time and a check of whether the difference between them does not exceed the predefined timer. Such a method has a positive impact on performance, as searching or updating the PFL requires only a single basic memory operation. However, the directly addressed table has one significant drawback: it does not ensure a full differentiation among the flows because they are indexed with identifiers calculated with a not-injective hash function. Theoretically, it would be possible to use a full concatenation of header fields as a table index without subsequent reduction with a hash function. However, a table indexed that way would require exabytes of storage. Therefore, lack of full differentiation among the flows is inevitable as far as a directly addressed approach is considered. As a result, collisions may appear, which result in the assignment of more than one flow to a particular flow identifier (table index). Therefore, a flow that is not present in the table may be recognized as a present one. Some flows that should not be protected may be protected instead. This would result in admission of more flows in congestion state, leading to the fair rate drop and longer congestion recovery time. Fortunately, flows that actually should be protected (are present in the table) always will be protected. The probability of collisions (and thus a potential fair rate drop level) may be successfully controlled by manipulating the PFL table size. Assuming usage of hash function giving a uniform distribution, the probability of collision can be calculated as a ratio of active flow number to the table size. For example, a 24-bit-wide index gives us a table with ~16 million slots. In order to achieve 1 percent maximum potential fair rate drop level, we should maintain no more than ~167,000 active flows in that case. Assuming usage of a 32-bit timestamp, such a table would need 67 Mbytes of RAM. For the purpose of the first tests, the directly addressed approach was chosen by the authors. A directly addressed flow list is much simpler to implement because it does not require implementation of complex data manipulation algorithms (like hash collision resolution or tree traversal) and garbage collection. In the first approach, assuming usage of a chained hash table for larger number of flows results in worse table operations performance (lookup, insert, and garbage collection) and higher memory usage. In the direct approach, table operations performance and memory usage are constant and do not depend on the number of active flows. Moreover, as far as the router may be used in laboratory networks, the number of simultaneous active flows will be low, so collisions will not be a problem. Thus, the choice of a direct addressing approach can be justified. The described issue is related to the concept of FAN, not to the implementation environment.

For packet scheduling, we decided to use the PDRR algorithm in our prototype router based on Click. This is because the complexity of this algorithm is low and the implementation process was the simplest among all three available algorithms. During the development process, two new Click elements were created. The first (PDRRSwitch) is responsible for the classification of flows into elastic or streaming types based on measured flow traffic characteristics, and for switching packets to proper outgoing queues. There is one priority queue for all router network interfaces where packets of streaming flows are stored. Moreover, each elastic flow is served by using a separate queue. The goal of the second, more complicated scheduling element (PDRRSched) is to serve outgoing queues according to the PDRR algorithm; that is, whenever the priority queue is not empty, it is served first and undergoes bufferless multiplexing while packets stored in the other queues share the remaining bandwidth in a fair manner. In addition, this second element is also responsible for performing outgoing link measurements. The values of two well-known FAN parameters, FR and PL, are estimated while packets exit queues, and then may be passed to the admission control block responsible for determining if a congestion state occurs in the link, which in turn influences new flow admission decisions.

Other issues connected with practical realization were observed during scheduling block implementation. As a consequence, the PDRR's assumption that each new flow (i.e., one that has no packets in its queue) arriving at the scheduling block should be treated with priority as a streaming one, some flows of evident elastic nature were served so quickly that their next packets arrived when the flow queue had already been emptied. Hence, they were treated as streaming flows and competed with actual streaming ones in the priority queue. Therefore, we propose some additional mechanism to prevent such behavior. A simple change was introduced, causing the ability for flows classified as elastic to be classified back as streaming only after some predefined timer expiration instead of classifying a flow as streaming right after its queue becomes empty. Introduction of the mechanism improves performance of streaming flows as they do not have to compete with flows that generally are classified as elastic but for some reason are being assigned to the priority queue from time to time (during one transmission). As later tests showed, it significantly improved streaming traffic performance in some specific scenarios. Timing values were adjusted based on empirical observations during the tests. The timer is fixed but configurable in the Click router configuration file. This mechanism can also easily be turned off in order to get back to pure FAN. Further research will be conducted on this mechanism to verify if it is a FAN or Click-specific issue. The source code of our implementation is available at [14].

Tests of the XP Router

After implementation of new Click elements and preparation of the router configuration file, it was possible to conduct the first tests of the developed prototype. Besides basic acceptance tests already performed during the implementation phase with use of built-in Click environment tools, some more advanced scenarios were tested in a network laboratory, including comparison of the XP router with a standard IP router provided by the Click environment to exclude general architecture differences.

Testing Methodology

The main goal of the tests was to show a working XP router and to confirm the usefulness of its functions. So far, only simulation results or numerical analyses presenting how the XP router works have been published. However, the implementation and tests in a real network, even if it is simple, allow some problems and effects to be noticed that are usually overlooked in simulation experiments. The best testing environment is a real network where traffic is generated by real users. If a device passes such tests, we can be almost sure that the tested device works properly and is scalable. However, it is very difficult and risky to test a prototype of a new device in a real network. On the other hand, numerical and simulation experiments usually show the basic functionality. They are, of course, very useful; however, the results may be obtained based on assumptions that are too general. As a result, some features of the proposed solution may not be noticed.

In this article, we present results of laboratory tests of the XP router. Our goal was to confirm the usefulness of this device in a small network with traffic generated by source nodes. Such tests allowed us to observe problems not noticed in simulation experiments performed before. First of all, we had to implement the PFL in a different way than originally proposed. Moreover, we had to modify the operations of the PDRR algorithm to eliminate too frequent changes of flow states. Our tests should be treated as a step between simulation experiments and performance analysis in real networks. They also set the option for further analysis of new mechanisms and algorithms proposed for the XP router in a laboratory environment.

We used prototypes implemented on relatively slow PCs. One of the features we were not able to present is scalability. However, to confirm it, dedicated platforms should be used. The hardware on which both routers were tested was based on a single-core processor platform with two external network cards. The results discussed below confirm our expectations that hardware resources influence the performance of the router. The most important factor is the CPU speed, which is directly related to router throughput. We used 1 Gb/s network cards, while the Click router maximum throughput varied between 100 and 300 Mb/s depending on the CPU used. Some additional equipment was also

Our tests should be treated as a step between the simulation experiments and performance analysis in real networks. They also set the option for further analysis of new mechanisms and algorithms proposed for the XP router in a laboratory environment.

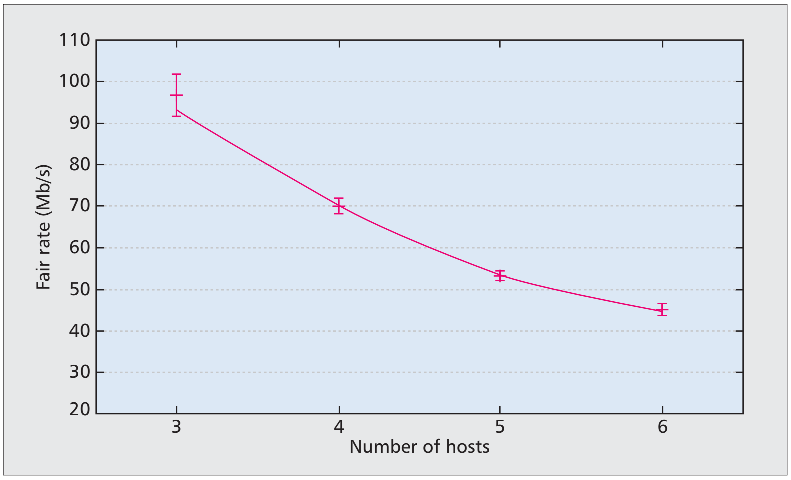

Figure 3. Fair rate measurements.

used, including network switches, VoIP PBX, IP phones, and computers serving as traffic sources/sinks.

Test Results

The first experiment was designed to analyze the functionality of the XP router, especially the PDRR scheduling block. One of the main goals of PDRR is to assign bandwidth to the elastic flows equally in a highly loaded link. To verify this feature, we prepared the laboratory environment. We used a simple topology consisting of hosts and server on opposite sides of the router. Tests were carried out in two configurations (with the standard IP router and the XP router), and in several scenarios, each of which assumed a different number of hosts sending data to the server (from one to six). Offered UDP traffic generated by hosts (one flow per host) and carried bandwidth on the server side were measured. In Table 1, the results from the selected scenario (with six hosts) are presented. All experiments were repeated at least 10 times. Ninety-five percent confidence intervals were calculated using the Student's t-distribution.

In this scenario, complete fairness of each flow is observed for the XP router. Independent of the amount of offered traffic, which varied from almost 100 Mb/s to over 400 Mb/s, the assigned bandwidth is almost the same for each flow. On the other hand, in the IP router case, flows are not balanced — they transmit at different rates. The IP router offers higher bandwidth to larger flows which is also consistent with our expectations, since it does not ensure fairness and usually tries to serve flows proportionally to incoming packets. Total traffic transmitted in a network with the XP router is lower than that observed for the IP router. The implementation of the XP router is more complex than that of a traditional IP router, and our device was not able to serve the same amount of traffic in both cases. More operations had to be performed in the XP router, consuming more processor power. As a result, a lower number of packets could be served in the same time period. The difference in total transmitted traffic is on the level of 10 percent. It may be treated as a first approximation of additional complexity added to the XP router in relation to the standard IP router. We can suppose that at least 10 percent of additional resources will be needed in the XP router to serve the same amount of traffic as in the standard IP router. We believe that the described differences in the total amount of transmitted traffic will not be observed when dedicated devices (with sufficient resources) are used.

In the second experiment, the operation of the admission control block and the method for estimating the values of the FR parameter were verified. The network environment was the same as in the previous experiment. We analyzed four scenarios, with three, four, five, and six hosts transmitting traffic. For each scenario, the FR measurement was made. Results of this test are shown in Fig. 3.

The obtained results fully agree with the FAN concept. With increasing occupancy of router resources (more traffic was generated), the value of FR decreases. This behavior fits the FR definition — the maximum rate that is or may be realized by a flow — and is consistent with our expectations. It is worth noting that small confidence intervals indicate the stability of the FR values. The obtained values of FR are comparable with bandwidth carried by a flow in a congested FAN link (in a case with six hosts, each source received around 47–49 Mb/s).

Finally, the XP router was also tested as a network device for use in a home or in a small company. For test purposes, an example network

Flow no.	Offered bit rate (Mb/s)	Carried bandwidth (Mb/s)		Carried bandwidth (%)	
		IP router	XP router	IP Router	XP Router
1	99.82 ± 0.14	21.78 ± 0.63	47.23 ± 1.57	21.82	47.32
2	158.07 ± 4.00	40.97 ± 6.87	47.95 ± 2.13	25.92	30.33
3	367.78 ± 24.66	64.43 ± 2.33	48.63 ± 0.91	17.52	13.22
4	365.59 ± 21.30	63.76 ± 2.52	49.05 ± 0.98	17.44	13.42
5	340.76 ± 8.27	62.91 ± 2.15	48.34 ± 1.02	18.46	14.19
6	403.76 ± 26.23	60.84 ± 3.76	47.25 ± 1.33	15.07	11.70

Table 1. Tests results of PDRR scheduling.

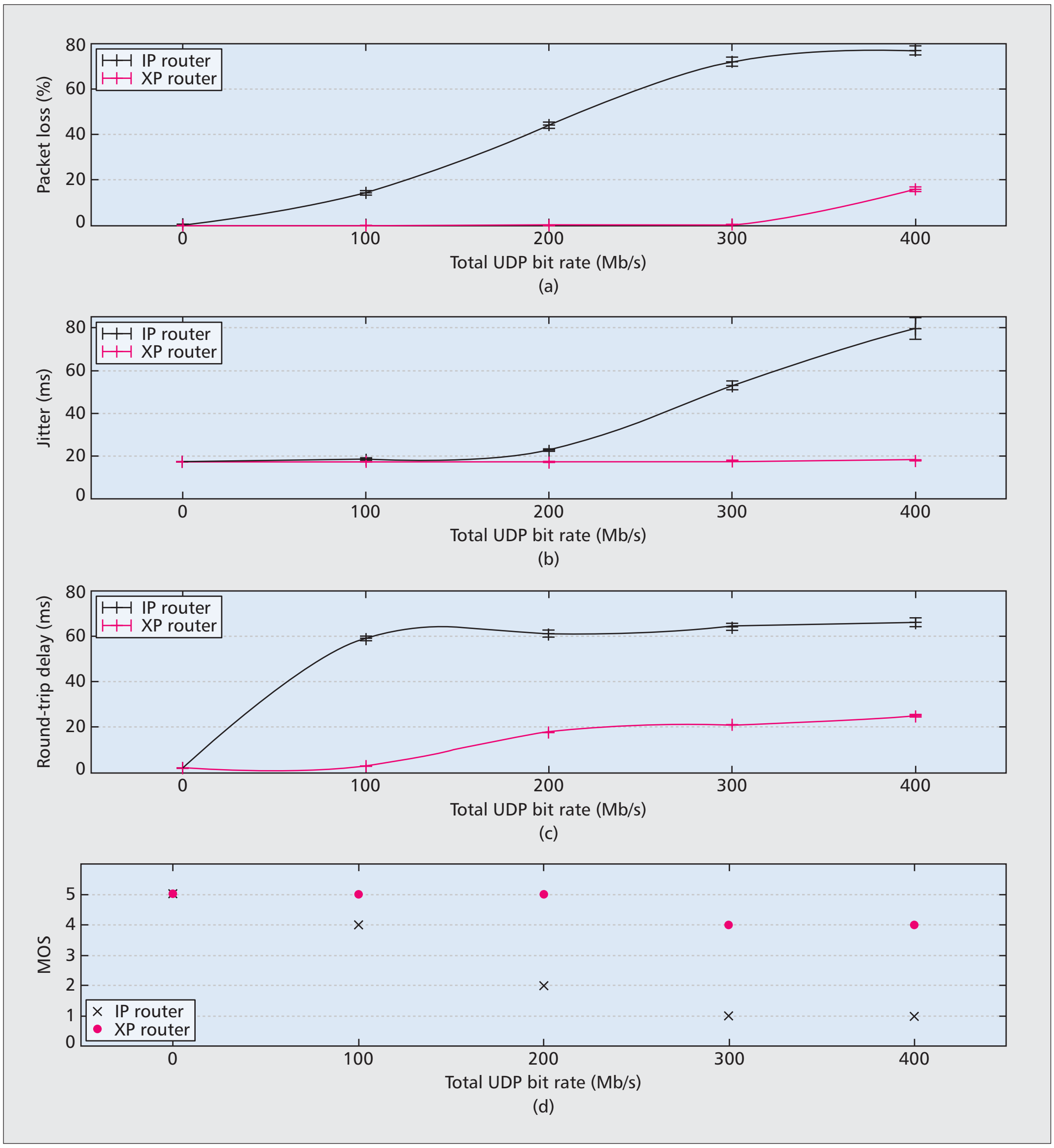

Figure 4. a) Variation of packet loss; b) jitter; c), round-trip delay; d) MOS in relation to UDP stream.

It is worth noting that small confidence intervals indicate the stability of the FR values. The obtained values of FR are comparable with bandwidth carried by a flow in a congested FAN link (in a case with six hosts, each source received around 47–49 Mb/s).

was prepared (Fig. 5). A few new elements were added to the previous environment to make it possible to set up VoIP connections.

Some Internet traffic, such as file downloads, as well as VoIP connection between the hosts, was transmitted in the analyzed network. An important component of this network was the VoIP PBX server, located on the Internet side. This was responsible for serving calls and measuring the values of the quality parameters. The aim of the tests was to check how the XP router handles the provision of call quality (VoIP calls were served as streaming flows) while other network users use the Internet intensively. There was one phone call ongoing while the volume of elastic traffic was increasing. Voice traffic bit rate was lower than 100 kb/s, while other traffic varied between 0 and 400 Mb/s. Analysis of router behavior was performed based on four criteria: packet loss (percent), jitter (milliseconds), round-trip delay (milliseconds), and mean opinion score (MOS) subjective rating. The values of the first three parameters were read directly from the PBX management interface. The MOS scale serves as a metric of quality of experience (QoE) [15]. The MOS scores were subjectively granted by several telephone users. Figure 4 shows a comparison of the previously mentioned parameters for both routers. As far as a conventional router is considered, in fact, there was no queuing policy at all as a single outgoing first-in first-out (FIFO) queue was used (default configuration provided with the Click environment). The purpose of the test was to analyze the impact of elastic traffic on VoIP call parameters. Thus, both traffic types were the objects of the measurements. Traffic volume was measured for UDP elastic flows, while the parameters such as packet loss, round-trip delay, jitter, and MOS were measured only for ongoing VoIP call.

The results of the tests show a significant advantage of the XP router over the conventional IP router. Each of the measured parameters indicates a huge difference between routers. Moreover, during tests of the IP router under

Tests in carrier-class networks could confirm the scalability of FAN. The authors believe that implementation will accelerate the research on flow-aware networks and open the possibility of implementing Cross-Protect router blocks in equipment used in real networks.

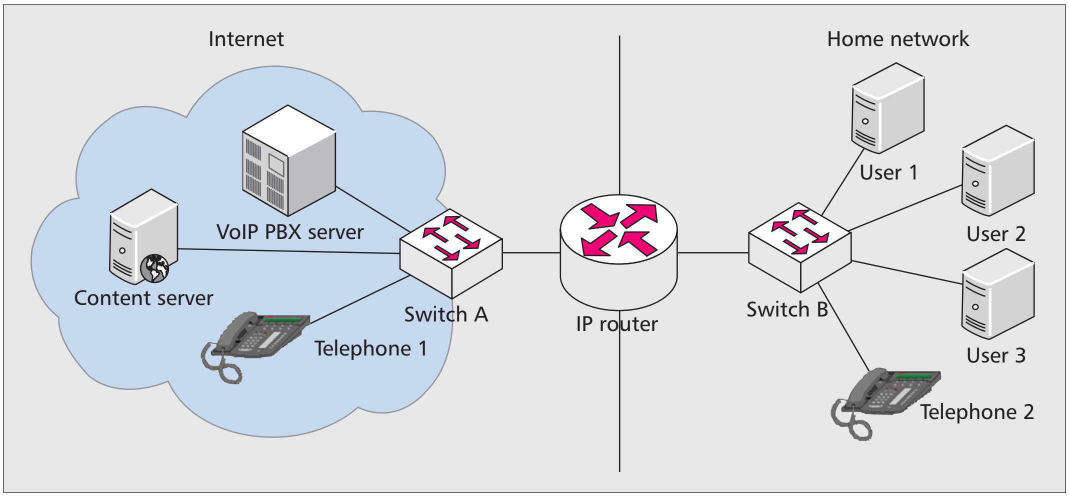

Figure 5. Network topology used for testing XP router in home/small corporate network environment.

maximum load, it was difficult to set up a VoIP connection. Because outgoing link bandwidth was set to 100 Mb/s, the results for the conventional router without any queuing policy are poor even at the level of 100 Mb/s of elastic traffic volume. Significant problems with call quality occurred during the transmission of 200 Mb/s between the router and switch B, and over this value, call setup was completely impossible. Under the same conditions, the XP router provided good quality of calls (4 in MOS) even if maximum load occurred. The values of the measured objective parameters confirm subjective users' rates. In our tests we generated only one priority (streaming) flow with low transmission speed. As a result, its influence on other traffic in an observed link was insignificant. Moreover, in FAN even elastic flows have guaranteed throughput, which is not observed for standard IP networks.

Conclusion

Flow-aware networking is a promising concept for the future Internet. It ensures good QoS for transmitted traffic and conforms to the net neutrality rules. The Click environment is an advanced and effective tool for prototyping of network devices.

The practical aspects of a prototype implementation of the FAN architecture's main element (the Cross-Protect router in the Click environment) are described in the article. The results of the first tests are also provided. It was necessary to add some new functionalities to the standard router implementation to develop the XP router empirical model. We also met several new challenges and problems in comparison to the simulation model analyzed by us previously. As a result, a few additional mechanisms or improvements have been proposed. Besides the basic tests verifying proper functioning of the implemented algorithms, some more advanced evaluation was conducted in the laboratory environment. Verification of two main FAN building block features, admission control and per-flow scheduling, was performed. Also, the means of delivering QoS in FAN was assessed. The test results are very promising and show a significant advantage of the XP architecture over the standard IP router in several areas.

Finally, we have to note that the conducted work proves the advantages of the FAN concept. Continuous development of the XP router device enables further FAN tests and experiments to be performed in the laboratory or even in real networks. In further work, already available promising simulation results of FAN may be verified in practice. Implementation may also be extended by several additional FAN mechanisms known in the literature. Tests in carrier-class networks could confirm the scalability of FAN. The authors believe that implementation will accelerate the research on flow-aware networks and open the possibility of implementing XP router blocks in equipment used in real networks.

Acknowledgment

The research was carried out with the support from the project "Implementation of Cross-Protect Router" no. LIDER/34/37/L-2/10/NCBiR/2011 founded by the National Centre for Research and Development in Poland under the LIDER programme.

References

[1] J. Roberts and S. S. Oueslati, "Quality of Service by Flow Aware Networking," *Philosophical Trans. the Royal Society of London*, vol. 358, Sept. 2000, pp. 2197–2207.

[2] D. Cuda *et al.*, "Building a Low-Energy Transparent Optical Wide Area Network With Multipaths," *J. Optical Commun. and Net.*, vol. 5, no. 1, 2013, pp. 56–67.

[3] J. Domżał *et al.*, "EFMP – A New Congestion Control Mechanism for Flow-Aware Networks," *Trans. Emerging Telecommun. Technologies*, 2013.

[4] G. Post and L. Noirie, "Method for Processing Data Packets in Flow-Aware Network Nodes," 2012, U.S. Patent 20,120,314,709.

[5] R. Wójcik and A. Jajszczyk, "Flow Oriented Approaches to QoS Assurance," *ACM Computing Surveys*, vol. 44, no. 1, 2012, pp. 5:1–5:37.

[6] A. Kortebi, S. Oueslati, and J. Roberts, "MBAC Algorithms for Streaming Flows in Cross-protect," *EuroNGI Wksp.*, Lund, Sweden, June 2004.

[7] R. Wojcik, J. Domżał, and A. Jajszczyk, "Predictive Flow-Aware Networks," *IEEE GLOBECOM '11*, Houston, TX, Dec. 2011.

[8] J. Domżał, N. Ansari, and A. Jajszczyk, "Congestion Control in Wireless Flow-Aware Networks," *IEEE ICC '11*, Kyoto, Japan, June 2011.

[9] J. Domżał, R. Wojcik, and A. Jajszczyk, "Per User Fairness in Flow-Aware Networks," *IEEE ICC* '12, Ottawa, Canada, June 2012.

[10] —, "Reliable Transmission in Flow-Aware Networks," *IEEE GLOBECOM '09*, Honolulu, HI, Nov.–Dec. 2009.

[11] J. Domżał *et al.*, "Admission Control Policies in Flow-Aware Networks," *11th Int'l. Conf. Transparent Optical Networks,* Azores, Portugal, July 2009, pp. 1–4.

[12] J. Domżał, "Intelligent Routing in Congested Approximate Flow-Aware Networks," *IEEE GLOBECOM '12*, Anaheim, CA, Dec. 2012.

[13] J. Domżał, R. Wojcik, and A. Jajszczyk, "QoS-Aware Net Neutrality," *INTERNET '09*, 2009, pp. 147–52.

[14] "Source Code of XP Router in Click," http://kt.agh.edu.pl/~jdomzal/fan-click-router.zip.

[15] R. Stankiewicz, P. Cholda, and A. Jajszczyk, "QoX: What Is It Really?," *IEEE Commun. Mag.*,vol. 49, no. 4, 2011, pp. 148–58.

Biographies

Jerzy Domżał received his M.S. and Ph.D. degrees in telecommunications from AGH University of Science and Technology, Kraków, Poland, in 2003 and 2009, respectively. Currently, he is an assistant professor in the Department of Telecommunications at AGH University of Science and Technology. He is especially interested in optical networks and services for future Internet. He is the author or co-author of many technical papers, five patent applications, and one book.

Jakub Dudek received his M.Sc. degree in telecommunications from AGH University of Science and Technology in 2012. His professional interests focus on flow-aware networking, network security, and TETRA systems. He is the co-author of several technical papers. He has been involved in national and European research projects. Currently, he is a senior system configuration engineer at Motorola Solutions.

Piotr Jurkiewicz is currently an M.Sc. student in the Department of Telecommunications at AGH University of Science and Technology. His research interests include software defined networking, flow-aware networking, adaptive routing, and fast packet processing. He is an open source software contributor. In 2013 he was a Google Summer of Code student at ON.LAB.

Łukasz Romański received his M.Sc. degree (with honors) in telecommunications from AGH University of Science and Technology in 2012. Currently, he works as a senior system test engineer at Motorola Solutions. He is the co-author of several technical papers and a patent application. He has participated in Polish and EU scientific projects. He is especially interested in quality of service, network performance testing, and network security issues.

Robert Wójcik received his Ph.D. (with honors) in telecommunications from AGH University of Science and Technology in 2011. Currently, he works as an assistant professor in the Department of Telecommunications of AGH. He is the co-author of five JCR journal papers, a number of conference papers, and four patent applications. He is a Technical Editor of *IEEE Communications Magazine*. He also serves as a Technical Program Committee member of several telecommunications conferences.