Flow-Aware Multi-Topology Adaptive Routing

Robert Wójcik, Jerzy Domżał, and Zbigniew Duliński

***Abstract*—Internet routing processes currently rely on protocols that were developed more than ten years ago. Today, we have far more computational power and memory at our disposal, and it is possible to take advantage of these resources in order to greatly increase the efficiency of routing protocols. Therefore, we propose a new approach to routing packets in IP-based networks: Flow-Aware Multi-Topology Adaptive Routing, or FAMTAR. FAMTAR combines flow-aware traffic management and an adaptive routing mechanism. A standard routing protocol is used to find the optimal path between two nodes in a network. FAMTAR makes it possible to automatically create additional paths when such demand occurs. Between two endpoints, the transmission may follow n different paths, where n is limited only by the topology of the network. In this letter, we compare FAMTAR to a classic routing protocol and demonstrate FAMTAR's superiority.**

***Index Terms*—Multipath, routing, flow, SDN.**

I. Introduction

As the Internet evolves, demand on resources is constantly increasing. These resources (bandwidth in particular) are often utilized inefficiently due to protocol constraints. The main source of inefficiency is that, by default, a routing protocol finds only the optimal path between two endpoints, and pushes all the traffic between the endpoints onto that path. This process continues even if the path becomes congested. Current routing protocols cannot deal with this problem. The most popular routing protocol, Open Shortest Path First (OSPF), was developed in 1998, and has hardly changed since then. As we have much more CPU power and RAM at our disposal today than 15 years ago, we are now able to use these resources and greatly increase the efficiency of routing protocols.

In this letter, we present a new approach to routing packets in IP networks: Flow-Aware Multi-Topology Adaptive Routing or FAMTAR. FAMTAR operates above the intra-domain routing protocol and can cooperate with every protocol. The idea is that a routing protocol is still responsible for finding the optimal path between two endpoints. In an uncongested network, all transmissions between those endpoints use this path. However, the optimal path may change according to the congestion status of links in the network. When the path becomes congested, all new flows are pushed to a new path, while flows which are already active remain on their primary path. Therefore, FAMTAR is responsible for using the optimal path between two endpoints provided by the routing protocol, and automatically finding new paths in case of congestion. Note that FAMTAR intends to modify the routing behavior inside the domain. It does not alter inter-domain routing.

FAMTAR is possible due to the popularity of flow-based traffic management and constantly increasing computational power in network devices. In FAMTAR, alternative routes are created and removed dynamically as needed. It is possible for transmissions to follow n different paths between two endpoints, where n is limited only by the topology of the network. FAMTAR therefore, allows the utilization of all available resources. If currently used paths become congested and another path exists, FAMTAR will find and use this path. Our experiments show that it is possible to significantly increase the amount of traffic sent in a network. In the examined topology, FAMTAR managed to double the amount of data that the network was able to carry in a given time. Moreover, traffic was transmitted with lower delays.

II. State-of-the-Art Solutions

Let us assume that in a given network there are three possible paths (A, B, and C) between a certain source-destination pair. A routing protocol chooses the path with the lowest cost as the designated route, say path A. Unless the network topology changes, all the traffic is pushed via path A. Even though paths B and C are available, they will never be used to transmit data for this pair. Therefore, some of the available resources will be wasted. Current devices can balance the load over multiple equal-cost paths. If path A and path B have the same calculated cost, the source can use both paths, in packet-by-packet round-robin, flow-by-flow, or using other routines. However, in large and complex networks it is very difficult to assure the existence of equal cost paths—and every modification of an interface cost changes the total cost of several paths.

Most routing protocols can only balance traffic on paths with equal cost. This is because it is difficult to provide unequal-cost load balancing without creating loops. However, with some additional functionality, unequal-cost load balancing is possible. For example, the proprietary Enhanced Interior Gateway Routing Protocol (EIGRP) routing protocol supports this. If enabled, the amount of traffic which is balanced to paths is proportional to their costs. This is inefficient, as suboptimal paths are utilized even when the optimal one could carry all the traffic. The solution which is proposed in this letter, uses the optimal path for as long as it is not congested. The second-best path is used when congestion is noticed, and then the third, and so on.

Network operators have a number of tools to manage traffic in order to achieve the functionality mentioned above. Multiprotocol Label Switching (MPLS) is the most popular example. MPLS allows the operator to manually create paths (tunnels) and assign certain flows/transmissions to those paths. A typical scenario would be that a routing protocol finds path A to be the designated path. The operator constantly monitors the amount

Manuscript received March 5, 2014; revised June 5, 2014; accepted June 23, 2014. Date of publication July 11, 2014; date of current version September 8, 2014. This work was supported by the Polish National Science Centre under Project DEC-2011/01/D/ST7/03131. The associate editor coordinating the review of this paper and approving it for publication was H. Luo. *(Corresponding author: Robert Wójcik.)*

R. Wójcik and J. Domżał are with AGH University of Science and Technology, 30-059 Kraków, Poland (e-mail: robert.wojcik@kt.agh.edu.pl).

Z. Duliński is with Jagiellonian University, 31-007 Kraków, Poland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LCOMM.2014.2334314

of traffic pushed via path A. If this amount threatens the quality of transmissions, the operator manually finds and creates a new path, say path B, and assigns some of the existing traffic onto it. This approach is currently widely used, and it works. However, there are several drawbacks. Firstly, the operations are not automatic and require human intervention. Secondly, in large and complex networks the existence of multiple paths and many criteria, conditions, parameters, etc., creates havoc. Currently, many major operators have their MPLS nodes configured with such complexity that they are almost afraid to change anything, for fear of impairing the network's operation.

Caspian Networks and, later, Anagran tried to provide flow-based treatment. In [1], it is shown that keeping flow state information is feasible. Moreover, Anagran created FR-1000, a router which provided flow-based treatment and could be used for high speed links. Anagran stores packet forwarding information inside flow tables, but unlike FAMTAR, this does not change according to network congestions. FAMTAR uses similar flow routing information to that used in Anagran, and combines it with routing adaptability to the current network congestion statuses.

A relatively new proposal for flow management was presented in [2]. In this solution, flows are classified and transmitted using multiple paths. A central manager decides which paths should be used for each flow. This proposal looks promising, however, it is complex and difficult to implement. The algorithm to select a path for a flow is run at the central node each time a new flow is accepted in the border router. Several metrics are taken into consideration to determine a path for a flow, e.g. network performance statistics including route message histograms. Moreover, the existence of the central manager may result in scalability and security problems in large networks. This is also the tendency of currently sound Software-Defined Networks (SDN). Unlike SDNs, FAMTAR does not require a central node to operate.

FAMTAR was developed to answer the aforementioned problems. This was possible thanks to the development of flow-aware traffic handling, such as Flow-Aware Networking (FAN) [3]. There were attempts to increase the efficiency of routing with the use of FAN, as e.g., presented in [4]. The technique of trunk reservation borrowed from the telephone network is proposed in this letter for route selection. It is assumed that a path for a flow is chosen based on the bandwidth it requires. Also, a simple intelligent routing for FAN is presented in [5], where it is assumed that only non-congested links are considered when forwarding packets. However, it is not specified how to inform all routers about congested links. FAMTAR provides a solution to that problem. Moreover, FAMTAR is a general idea, not specific to FAN networks.

III. FAMTAR

FAMTAR uses the currently popular network traffic management which is based on the concept of flows [6]. Although the idea of a "flow" is defined differently in various literatures, it always means a stream of packets belonging to a certain transmission between two end users. For example, the FAN architecture defines a flow as in [7]: "the flight of datagrams, localized in time and space, and having the same values in certain packet header fields." These packet header fields are the so called "5-tuple": source and destination IP addresses, source and destination port numbers, and the name of the transport layer protocol used for transmission, e.g., TCP or UDP.

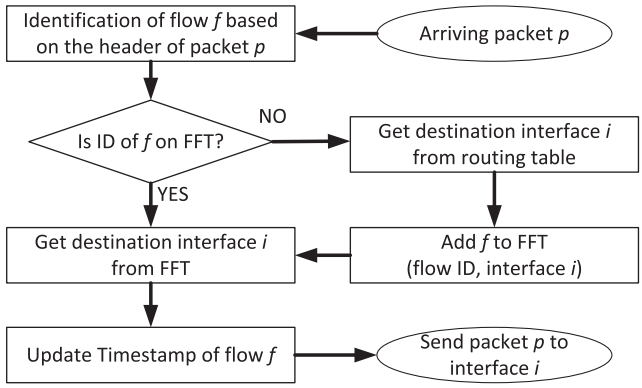

Fig. 1. Packet processing in FAMTAR.

The processing of a packet in a FAMTAR router is presented in Fig. 1. FAMTAR uses an additional component to the standard router functionality: a Flow Forwarding Table (FFT). When a packet of a new flow arrives on a router, the ID of a flow (represented by a hash of the 5-tuple) is computed. Next, it is checked whether this flow is present in the FFT list. If it is, the router acquires the outgoing interface from the FFT, updates the Timestamp of a flow to the current time, and forwards the packet. In this procedure, the routing table is not consulted. If a flow is not present in the FFT list (because it is a new flow), the respective outgoing interface is derived normally, i.e., from the routing table. Afterwards, the flow is added to the FFT along with the interface number to which the packets of this flow should be forwarded.

The FFT is basically used to realize the routing tasks, as for flows that are present on the FFT, the routing table is not needed. The FFT stores information about the time at which the last packet of a flow appeared. This is necessary to erase inactive flows and keep the list as small as possible. Such an approach may appear at first glance to have problems with scalability, but a similar flow treatment was analyzed in [8] and proved to be scalable. Moreover, currently popular OpenFlow-based devices implement a similar table and functionality.

In FAMTAR, the process of determining the optimal paths remains for the traditional routing protocol. When there is no congestion, the whole network operates in a standard, classic way. When a state close to congestion is noticed on one of the links, the corresponding router sets the cost of this link to a predefined high value. From this moment, this link is perceived by FAMTAR as congested. The new cost appears as a change in the routing protocol, which disseminates this information as a standard topology change message. Upon receiving this information, routers compute new routes. The path containing the congested link will have such a high total cost that it will not be optimal. New routes will include the congested links only when no other path is possible. The congested link still forwards all the flows which were active before the congestion was noticed. However, new transmissions do not appear, because this link is no longer present in current designated paths. After a while, when the congestion on this link stops, the original cost of the link is restored and optimal paths may contain this link again. Note that FAMTAR requires a router to detect congestion

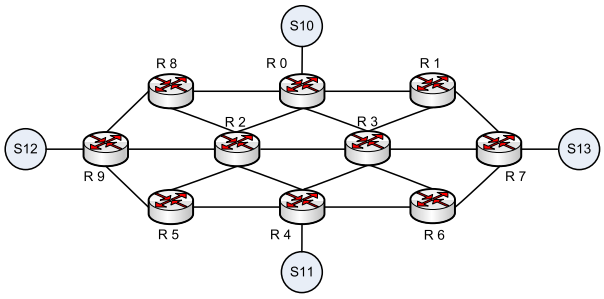

Fig. 2. Simulation topology.

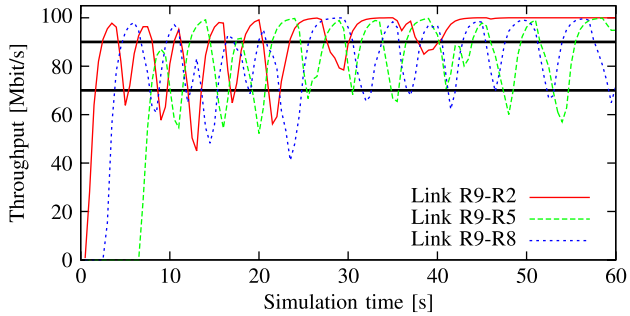

Fig. 3. Core links' throughput.

on one of its links. The method to determine the congestion is not specified, therefore, any congestion indicator can be used (e.g., link load, queue occupancy, packet queuing delay, and so on). Also, the matter of calculating new optimal routes is not modified by FAMTAR.

Say that path A is chosen by the routing protocol. All packets are forwarded to path A until congestion appears. The congestion is noticed on one of the links in Path A. The cost of this link is increased and the routing protocol now sees path B as optimal. Path B is therefore installed in the routing table. The ongoing flows do not observe any difference, as they are still being forwarded to path A (their designated interface is taken from the FFT list, rather than from the current routing table). New flows, on the other hand, receive the outgoing interface from the routing table, and this interface is inserted into the FFT. When congestion is noticed on path B, path C takes over, and so on. When congestion on path A stops, new flows start to use path A, as normally, this path is more desirable than paths B and C.

FAMTAR provides just the possibility to use multiple paths when such request appears. It does not mean that those paths need to be utilized at all costs. A network operator can easily consider only certain top paths, or paths with the certain cost ratio. Moreover, it is very easy to implement admission control functionality, so that routers do not accept new flows on congested links. Also, it is possible to combine FAMTAR with existing or new QoS mechanisms. For example, certain traffic can be prohibited from choosing alternative paths and be pushed to the primary path all the time. The possibilities are endless.

IV. Evaluation

FAMTAR was evaluated by $ns-2$ simulations using the topology in Fig. 2. The results are divided into scenarios.

Scenario I: In the first scenario, traffic was sent only from S12 to S13. We generated 10 000 TCP flows; the volume was modeled using the Pareto distribution with the mean flow size of 10 MB and the shape factor of 1.5. The packet size was set to 1000 bytes and the flows were generated with interarrival times modeled with the exponential distribution (mean interarrival time: 25 ms). We used the OSPF routing protocol. The simulation duration was set to 250 seconds and the warm-up time was 30 seconds. The capacity of access links (S12-R9, S10-R0, S13-R7, S11-R4) was set to 1 Gbit/s and the capacity of core links was 100 Mbit/s.

We observed the throughput on links between routers R9-R2, R9-R8 and R9-R5 to demonstrate how FAMTAR works. The results are presented in Fig. 3. Two thresholds were defined, $Th_{\min}$ and $Th_{\max}$. If the throughput of a link is greater than $Th_{\max}$, the link is congested and its cost is changed to the maximum possible value allowed in the routing protocol. The congested link's cost returns to the original value when throughput of this link drops below $Th_{\min}$. The gap between thresholds, referred to as threshold width, ensures a reasonable number of cost changes, thereby, reducing the number of routing protocol path computation processes. Evaluation presented in Scenario II shows that the threshold width of $0.2 * C$ gives satisfactory results.

Ensuring that link utilization is below 90% is critical for good transmission quality of flows, as higher utilizations usually provide heavy packet losses and delays. Therefore, we set those values to $0.7 * C$ and $0.9 * C$, accordingly, where C is the link capacity. In our simulations, those values yielded best results. However, these might not be appropriate for all networks and any operator adopting FAMTAR can set them suitably. Moreover, these thresholds do not need to be static; they can be adjusted dynamically according to current requirements imposed on the operator.

In Fig. 3 we see that, at the beginning, flows are accepted on link R9-R2. When the throughput of this link exceeds $Th_{\max}$, new flows are accepted on link R9-R8. Similarly, when this link becomes congested the packets of new flows are forwarded to link R9-R5. However, when the congestion on link R9-R2 stops, the link begins to accept new flows again. Originally, link R9-R2 was selected by the routing protocol, which means that it had the lowest cost. If all three links are congested, R9-R2 is still the best choice. As a result, link R9-R2 becomes overloaded from approximately 45 seconds until the end of the experiment. Once all possible paths are congested, it could be a good practice to block all new traffic in order not to degrade the performance of ongoing flows. This can be easily implemented in FAMTAR. A simple admission control block which does not allow new flows when a link is congested would suffice.

Scenario II: In the second scenario, traffic was sent from all sources. Each source node was sending traffic to randomly chosen destination node. Each source node generated 10 000 flows with the same parameters as in the previous case. The rest of the simulation parameters were identical. The simulation experiment was repeated in order to obtain statistically credible results. 95% confidence intervals were calculated using the Student's t-distribution. We observed the total amount of traffic transmitted in the network during the observed time period in relation to the threshold widths. The results are presented in Fig. 4.

We can see that the amount of transmitted traffic in the network decreases with the increasing threshold width. Moreover,

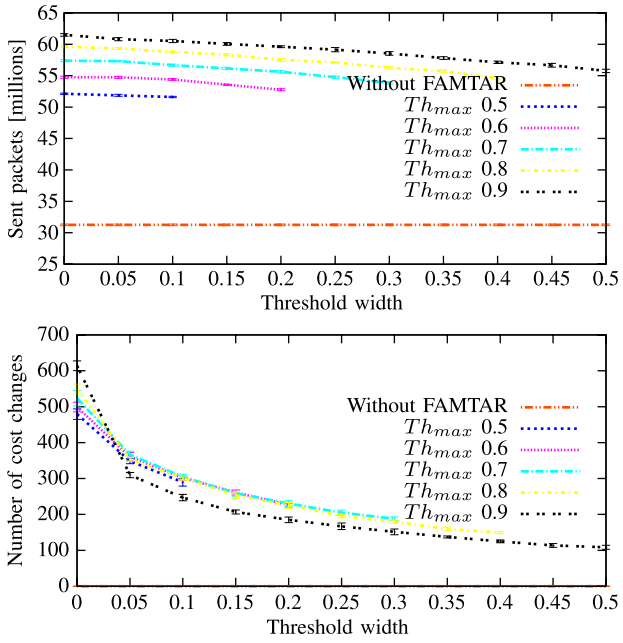

Fig. 4. Transmitted data and link cost changes in relation to threshold width.

TABLE I
THE ADVANTAGES OF FAMTAR OVER STANDARD IP ROUTING

Observed parameter	std. network	FAMTAR
Received packets [millions]	27.58 ± 0.22	55.35 ± 0.46
Received to sent packets ratio	0.88 ± 0.01	0.93 ± 0.01
Received traffic [GB]	14.79 ± 0.15	29.08 ± 0.20
Received to sent data ratio	0.90	0.95
Mean packet delay [ms]	28.17 ± 0.20	23.04 ± 0.72
Mean hop count	5.02 ± 0.01	5.42 ± 0.20
Min / max hop count	5 / 6	5 / 9
Flow mean trans. time [s]	94.52 ± 0.98	74.88 ± 1.84
Link cost changes	0	189.4 ± 21.17

with the higher value of $Th_{\max}$ we are able to send more traffic. Most important, however, is the fact that in each case with the FAMTAR mechanism, we are able to send much more data than in a standard network.

The best results are observed for $Th_{\min} = Th_{\max} = 0.9$. However, we have to be aware that decreasing threshold width dramatically increases the number of routing table changes. This situation is illustrated in the bottom plot of Fig. 4. We can see that setting a threshold width of $0.2 * C$ allows for significant reduction of link cost changes in a network. The further increase of threshold width does not result in significant reduction of link cost changes.

For $Th_{\max} = 0.9$ and $Th_{\min} = 0.7$ we also provided more detailed analysis, observing the amount of traffic sent in the network, the number of link cost changes, delays, mean transmission time of flows and other factors. The results are presented in Table I. We can see that in the analyzed network, FAMTAR allows the transmission of almost twice the amount of traffic. Moreover, the ratio of the number of received packets to the number of sent packets is higher and the mean packet delay is lower. It is worth noting that mean packet delay is lower in the network with FAMTAR, even though the mean number of hops is increased.

V. CONCLUSION AND FUTURE STUDIES

In this letter, we advocate a new approach to routing packets. It was shown that FAMTAR allows to increase the amount of traffic sent in a network, significantly. Moreover, traffic is transmitted with lower delays and with a higher ratio of received to sent data.

There are two operational costs of FAMTAR. One is the necessity to maintain and use the FFT. However, in the era of SDNs, OpenFlow etc., this requirement is nothing novel. Another is the necessity of changing the costs of links according to congestions observed in a network. In our case, the routing table of R9 changed almost 200 times during 250 seconds of network operation. This seems excessive, however, it has only a limited impact on performance. Firstly, it does not degrade packet forwarding process of all ongoing flows as they are forwarded according to FFT and for this process the routing table is not used. Secondly, in current routers, routing and packet forwarding processes are separated in such a way that routing processes do not degrade packet forwarding.

FAMTAR suffers from transient states and failures which can cause loops. If this happens, there needs to be a way of getting out of loops since a routing protocol does not impact ongoing flows. We are currently investigating a method to solve the problem. The idea is to check if a packet's Time to Live (TTL) value matches the one stored in the FFT list once the flow was admitted. If there are loops, TTL mismatches will detect them, and proper actions need to follow. Initial results look promising, but more research is necessary.

REFERENCES

[1] L. G. Roberts, "The next generation of IP—Flow routing," in *Proc. Int. Conf. SSGRR*, L'Aquila, Italy, Jul. 2003, pp. 1–15.
[2] I. Rubin and R. Zhang, "Max-min utility fair flow management for networks with route diversity," *Int. J. Netw. Manage.*, vol. 20, no. 6, pp. 361–381, Nov./Dec. 2010.
[3] J. Roberts and S. Oueslati, "Quality of service by flow aware networking," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 358, no. 1733, pp. 2197–2207, Aug. 2000.
[4] S. Oueslati and J. Roberts, "Comparing flow-aware and flow-oblivious adaptive routing," in *Proc. 40th Annu. CISS*, Baltimore, MD, USA, Mar. 2007, pp. 655–660.
[5] J. Domzal, "Intelligent routing in congested approximate flow-aware networks," in *Proc. IEEE GLOBECOM*, 2012, pp. 1751–1756.
[6] R. Wójcik and A. Jajszczyk, "Flow oriented approaches to QoS assurance," *ACM Comput. Surv.*, vol. 44, no. 1, pp. 5:1–5:37, Jan. 2012.
[7] A. Kortebi, S. Oueslati, and J. W. Roberts, "Cross-protect: Implicit service differentiation and admission control," in *Proc. HPSR*, Phoenix, AZ, USA, 2004, pp. 56–60.
[8] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts, "On the scalability of fair queuing," in *Proc. ACM HotNets III*, San Diego, CA, USA, Nov. 2004, pp. 1–6.