All-Path Family: Scalability and Load Balancing for Bridging Networks

Elisa Rojas
University of Alcala (UAH)
Alcalá de Henares, Spain
elisa.rojas.sanchez@gmail.com

Guillermo Ibáñez
University of Alcala (UAH)
Alcalá de Henares, Spain
guillermo.ibanez@uah.es

***Abstract*—This paper presents a study of the *All-Path* family, a new family of switching protocols that gets its name from the fact that all paths of the network are simultaneously explored with a broadcast frame forwarded over all network links to find a path or set a multicast tree. ARP-Path is the first protocol of this family, which upon arrival of a new flow to the network, it explores every possible path reaching from source to destination host and selects the lowest latency path at that moment, by snooping the ARP protocol dialog (request and reply messages). With the same principle, two new types of All-Path protocols evolve from ARP-Path: Flow-Path and Bridge-Path, flow-based and bridge-based respectively, instead of source address-based. While preserving the main advantages of ARP-Path: zero configuration shortest path bridging exploiting the full network topology, Flow-Path has the advantages of full independence of flows at the time of path creation and guarantees path symmetry (congruency) and increased path diversity, and Bridge-Path increases scalability by reducing forwarding table entries at core bridges. In the study we will compare the characteristics of each protocol and the convenience of using each one depending on the topology and the type of traffic.**

***Index Terms*—Ethernet, Switching, Bridging, Routing bridges, Shortest Path Bridging, Data Centers**

I. Introduction

Ethernet switched networks offer important performance and cost advantages for local, campus and metropolitan networks, with a high compatibility between elements, and a simpler configuration than IP. But the Spanning Tree Protocol (STP/RSTP) [1] severely limits the network size and performance by blocking redundant links to prevent loops, thus limiting infrastructure utilization and increasing latency, while SPB [2] and TRILL RBridges [3] still need some complex configuration since they are based on a link-state protocol. In this situation, simple, zero configuration protocols that remove the limitations of RSTP and, at the same time, allow scaling Ethernet to deal with bigger single IP subnets with no link-state protocols, might become the key to boost its deployment on campus, data center and even enterprise networks.

ARP-Path protocol has recently emerged as a shortest path -more precisely, low latency path- bridging alternative [4]. It is based on evolved bridging mechanisms that take advantage of the information conveyed by the ARP protocol message dialog to construct the forwarding table. The high degree of path diversity (load sensitive path selection) achieved by the ARP-Path protocol produces excellent results both in terms of latency and overall throughput [4], but there are some special scenarios where overall performance may not be optimal: networks with very asymmetric traffic patterns or where state store should be kept low.

II. All-Path Family

To understand the All-Path family is important to understand first the ARP-Path protocol [4]. ARP-Path takes advantage of the fact that, previous to any communication, an ARP will be emitted. The ARP protocol *request* and *reply* messages are snooped at ARP-Path bridges to select low latency paths with zero configuration and full transparency to both hosts and switches. In Fig.1, there is an example in which the ARP Request is snooped in every bridge to learn A's address and, at the same time, late copies (copies of the ARP Request that arrive at a bridge that already learnt A's address) are discarded to avoid loops.

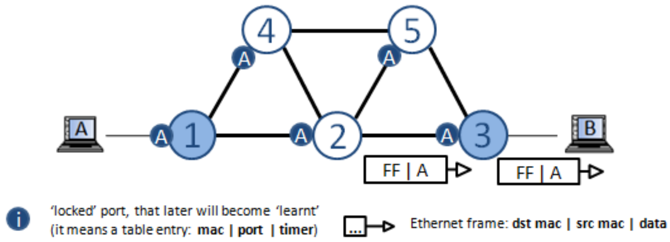

Fig. 1. Learning of path to A by snooping the ARP Request emitted from A

Flow-Path follows the same steps, but the path created is both for source and destination of the communication, i.e. the flow, while in Fig.1 the path learnt for A is not only valid for the destination B, but for any other host willing to communicate with A. And in the case of Bridge-Path, bridges learnt the identity of edge bridges instead of the hosts, i.e. in Fig.1 they would learn 1 (A's edge bridge) instead of A. In order to learn that identity, it should be propagated within the ARP messages through the network. To do so without modifying the ARP message at all, there are three alternatives: using the VLAN tag (ARP-PathV), encapsulating the frame with MAC-in-MAC (ARP-PathM) or translating the host address into a hierarchical address in which one field indicates the bridge identity (Path-Moose).

III. Analysis of Scalability VS Load Balancing

From their definitions, it is easy to see that Flow-Path will lead to more capability of load balancing, because it creates many different paths for a single host, followed by ARP-Path and finally Path-Bridge, which creates a single path for a group of hosts connected to a bridge. Nevertheless, the state-stored cost is also much higher for Flow-Path and the lowest for Bridge-Path, being this also important for the scalability of the network. Thus, in this section we analyze the suitability of each protocol for different topologies, in order to guarantee the optimal load balancing and table sizing.

The convenience of using a protocol more capable of balancing load is known by measuring the available paths in a topology and comparing that number with the possible paths that could be created, i.e. average active flows (Flow-Path), average active hosts (ARP-Path) and average active edge bridges (Bridge-Path). Notice that the number of possible paths is theoretical and can be bigger than the number of available paths. While in order to measure the scalability, it is necessary to measure the average number of table entries in the topology for each protocol:

$$T_{FP} = F_U * b = H(H\text{-}1) * b \qquad (1)$$

$$T_{AP} = H * (b+Le) \qquad (2)$$

$$T_{BP} = B_E * (b+Le) \qquad (3)$$

Being T_{FP} the average number of table entries for Flow-Path, T_{AP} for ARP-Path and T_{BP} for Bridge-Path, while the rest average numbers in the equations are F_U: unidirectional active flows, H: active hosts, B_E: active edge bridges, b: number of bridges conforming the average path, Le: number of extra bridges that also share the path.

Notice that, in the previous equations, the entries are multiple of the number of flows (which at the same time are multiple of almost the square of the number of hosts) for Flow-Path, multiple of the hosts for ARP-Path and multiple of the edge bridges for Bridge-Path. For a fixed topology the edge bridges are a constant, while the average active hosts depend on the type of traffic. They are also multiple of b or $(b+Le)$, the difference is that Flow-Path generates a single path b for a flow, while ARP-Path and Bridge-Path share the path for a source (host or bridge) with multiple destinations, i.e. they create a tree from the source to any destination, in which b is the average path and Le the branches of the tree (a value that will depend on the shape of the network). Therefore, the state store will not only depend on the number of active hosts, but also on the type of topology.

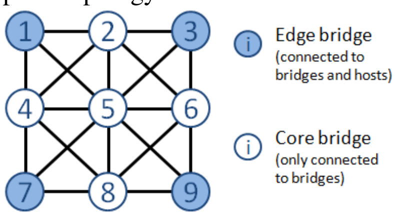

Fig. 2. Grid topology of size 3x3

Let us consider the grid topology shown in Fig.2. with four edge bridges, one at each corner and size *n x n* (where n=3 in the figure). We analyze the convenience of using each protocol, for the optimal load balancing and tables entries, in function of the size *n* and the average number of active hosts in the topology *H*.

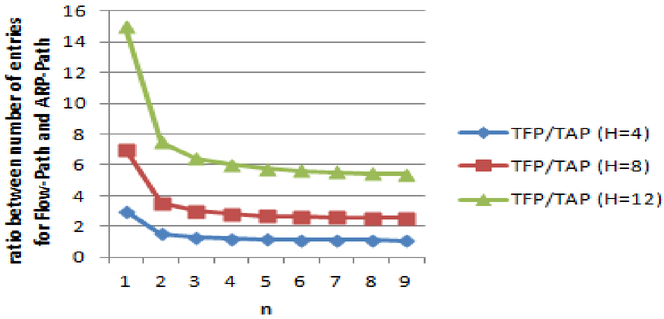

Fig. 3. Ratio between the number of table entries for Flow-Path and ARP-Path in function of *n* (size) and *H* (average active hosts in the topology).

In Fig. 3 we can see that the ratio between T_{FP} and T_{AP} decreases as the size increases, up to a constant lower than (*H-1*), due to the effect of *Le*. In Fig. 4 we can see that for *H*=4 (one host at each edge bridge), all protocols have the same load balancing, and increases with *H*, for *H*=8 Flow-Path is optimal with *n*=4, but for *H*=12 Flow-Path offers more possible paths than the network and ARP-Path equals the network for *n*=3, so maybe ARP-Path is the optimal solution.

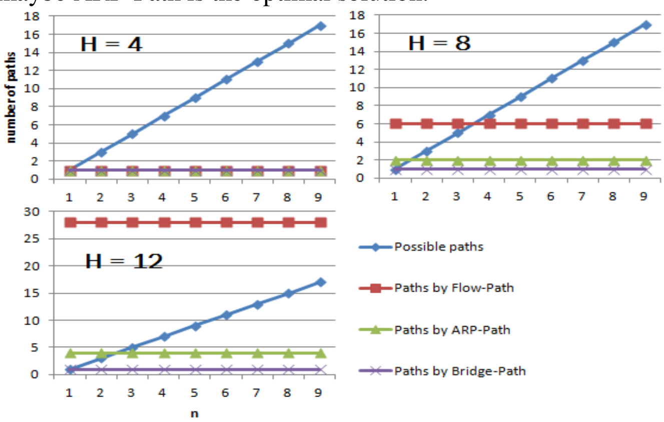

Fig. 4. Available paths (blue) and paths created by each protocol (red: Flow-Path, green: ARP-Path, purple: Bridge-Path) in function of *n* and *H*.

IV. Conclusion

The All-Path family has many optimal bridging solutions considering load balancing and scalability for different topologies and traffic patterns, but it is necessary to analyze which is the best. We show an example for a grid topology. Further results (OMNeT++ simulations) are not given due to lack of space.

References

[1] IEEE 802.1D-2004 - MAC Bridges: http://www.ieee802.org/1/pages/802.1D-2003.html

[2] IEEE 802.1aq – Shortest Path Bridging: http://www.ieee802.org/1/pages/802.1aq.html

[3] RFC 6325 – Routing Bridges (RBridges): Base Protocol Specification: http://tools.ietf.org/html/rfc6325

[4] G. Ibanez et al. "ARP Path: ARP-based Shortest Path Bridges". IEEE Communication Letters July 2011