A Simple, Zero-configuration, Low Latency, Bridging Protocol

Guillermo Ibáñez[1], Jad Naous[2], Elisa Rojas[1], Diego Rivera[1], Juan A. Carral[1], José M. Arco[1]

1 University of Alcala,, Madrid Spain 2 Stanford University, Stanford, CA USA

***Abstract*—This paper describes a demo for a new type of bridges, ARP-Path bridges. These ARP-based Ethernet Switches rely on the race between ARP Request packets flooded over all links, to discover the minimum latency path to the destination host. The protocol uses all links, is loop free, uses the standard Ethernet frame format, is fully transparent to hosts and neither needs a spanning tree protocol to prevent loops nor a links state protocol to obtain minimum latency paths. Implementations in Linux and Openflow on NetFPGA show inherent robustness and fast reconfiguration. Simulation results show throughput and delay performance superior to the Spanning Tree Protocol and similar to shortest path routing, with lower complexity.**

***Index Terms*—Ethernet, Routing bridges, Shortest Path Bridges, Spanning Tree**

I. Introduction

Ethernet switched networks offer important advantages in terms of price/performance ratio, compatibility and simple configuration without the need of IP addresses administration. But the spanning tree protocol (STP) [1] limits the performance and size of Ethernet networks. Current standards proposals, such as Shortest Path Bridges (SPB) [2] and Routing Bridges [3] rely on a link-state routing protocol, which operates at layer two, to obtain shortest path routes and build trees rooted at bridges. However, they have significant complexity both in terms of computation and control message exchange and need additional loop control mechanisms.

In this paper, we propose a demo for ARP-Path Ethernet Switching, recently proposed in [4] as Fast-Path. ARP-Path is a simple, zero-configuration, minimum latency protocol suitable for metro, campus, enterprise, and data center networks that enables the use of all available links without routing computations or a spanning tree. Our simulations show 2 to 5 time higher infrastructure utilization [4], higher availability than RSTP and throughput and delay similar to shortest path routing at a fraction of its complexity.

II. ARP-Path Protocol

A. *ARP-Path Set up*

The ARP-Path protocol relies on the race between flooded ARP requests to establish the fastest path.

1) *ARP-Path Discovery (ARP Request).Fig.1*

When host S wants to send an IP packet over Ethernet to host D by IP address, it needs D's MAC address. If this mapping of IP address to MAC address is not in its ARP cache, D broadcasts an ARP Request for D's MAC address (shown as B in fig.1a) . Ingress bridge 2 receives the frame from S and temporarily associates (*locks*) the global MAC address of S to the ingress port. Further broadcast frames from S arriving to other input ports of bridge 2 will be discarded as late frames from that source. S's address is now in a locked state and bridge 2 broadcasts B on all other ports (fig.1 a). Bridges 1 and 3 behave similarly, locking S's address to B's ingress port and broadcasting B over all other ports, thus sending duplicate copies to each other. Because these frames arrive at a different port from the one already locked to S, they are discarded (fig.1 b). In turn, bridges 4 and 5 process B the same way finally delivering B to the destination host D. There is now a chain of bridges each with a port locked to S's MAC address forming a temporary path between S and D (fig.1 c).

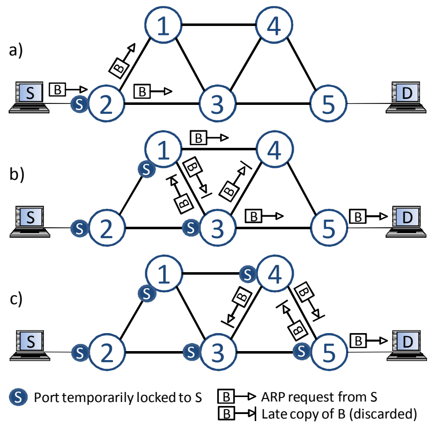


Figure 1. ARP-Path discovery from host S to host D. The small bubbles on the links show which switch port locked S's address.

2) *ARP-Path Confirmation (ARP Reply)*

The path is confirmed in the reverse direction (i.e. from D to S) when Host D sends the ARP Reply to host S in a unicast frame U that traverses the same path and confirms the learned addresses. The confirmation mechanism ensures that the ARP-Path is symmetric. Path symmetry is required to prevent path oscillations. Specific priority mechanisms are used to ensure the uniqueness of the confirmed path in special situations like simultaneous ARPs exchanged by two hosts in opposite directions.

B. *ARP-Path Repair*

The method for path repair when a unicast address arrives at a bridge where the destination MAC address has expired is described in detail in [4]. It uses Path_Fail, Path_Request and Path_confirm packets between the affected bridges to restore the complete path between source and destination bridges by emulating the ARP exchange.

III. Advantages

The protocol has several important advantages over other protocols that that explicitly build routes.

- Minimum Latency. The selected path is the minimum latency path as found by the ARP Request message.
- Zero configuration. There is no need to configure anything on hosts and bridges
- Simplicity.
- Reduced broadcast. ARP-Path increases ARP broadcast frames by a factor *(d-1)* [4], where *d* is the average node degree because it uses all network links. But implementing ARP proxy at edge ARP-Path bridges is functionally simple (adding IP to the learned MACs) and provides a drastic reduction of broadcast and ensures stable paths in network. The source bridge of every host, once the path is found by the first ARP Request, may respond to further requests to hosts and avoid repetitive requests and even optionally refresh MAC addresses close to expiration [5].The two variants of ARP proxies in broadcast reduction have been studied in depth in [5]. If ARP Proxy function is implemented in all ARP-Path bridges, the effective ARP Request rate will be the result of multiplying the original rate by the miss rate of the proxies.

IV. Evaluation

We present here only a short summary of the results of our evaluation shown in [4]. We also provide performance measures obtained from software simulations.

A. *Simulation results*

We implemented an ARP-Path Ethernet switch simulator in Omnet [6]. Here, we compare the performance of ARP-Path, shortest path routing, and STP, focusing on data flow performance across flat mesh topologies like the pan-European core reference network [7].

1) *Latency and network availability.*

ARP-Path is a reactive protocol but it does not increase network latency in normal operation because hosts need ARP resolution to complete before they can communicate anyway. When path repair is needed because an address has expired in a switch due to reconfiguration it takes a maximum of two round trip times (RTT) across the network of bridges (inspecting the frame at each traversed switch) to rebuild the path (assuming the method in II.B is used). Only the paths affected by a port or link failure will need repair, all other paths remain valid, unlike spanning tree where all MAC addresses must be flushed after reconfiguration because association to ports could change. Latency values obtained with the Omnet simulator are as follows: for a typical enterprise network [4] with 4 core and 10 access switches (5-hop path length) the path restoration time obtained in Omnet ranges from 5,25 to 8 msec depending on location of the link failed.

2) *Throughput*

Simulations at [4] (see Fig. 2) show the average link load obtained at the most loaded link *(bottleneck link)* of a typical campus network versus the percentage load (relative to maximum link capacity) at a client host link (both client and network links have the same capacity of 100 Mbit/s). As expected, using spanning tree produces the worst results because the routes are concentrated around the root bridge.

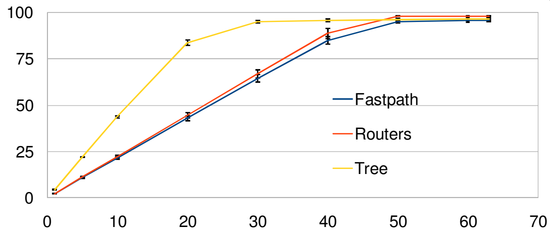


Figure 2. Throughput comparison, % of most loaded link versus % of traffic load at the sending host link. Simulated links at 100 Mbit/s.

B. *THE DEMO*

The objective of the demonstration is to show the performance, reconfiguration speed and robustness of the ARP-Path transparent bridge concept at 100 Mbit/s and 1 Gbit/s wired networks, and the use of all infrastructure links, without the spanning tree protocol or any ancillary routing protocol at layer two or three. Demo comprises a local demo at 100 Mbit/s at the conference venue with 3 ARP-Path Linux boxes (Soekris boards) and a video (or remote execution whenever possible) of a demo at 1 Gbit/s with three or four switches made with NetFPGA boards running Openflow. We demonstrate the compatibility with real networks, fast reconfiguration after link failure and absence of broadcast storms or other infinite loops. No broadcast loops occur even when two ports of the same bridge are connected to each other.

1) *100 Mbit/s. ARP Path Switches Linux Implementation*

In this demo the ARP-Path bridge protocol has been implemented on a Linux 2.6 kernel and operates in kernel and user space using *ebtables* [10]. The demo scenario is shown in figs. 3and 4. Three Linux boxes operate as three ARP-Path switches (labelled *F*) fully connected in a triangle. Two of these switches are connected to standard hosts (*H, S*), and the third switch is connected to the Internet through a wired Fast Ethernet connection or through a host acting as wired-to-wireless gateway. Path establishment between H and S is continuously monitored with repetitive pings. The first ping takes more time because path has to be set up at switches and Linux bridges execute it partially in user space. Disconnecting a link of the path in use activates path reconfiguration, with a path repair time of around 1100 milliseconds after link failure. The new selected path is, obviously, the alternative and longer two-hop path via the third bridge, with a bit longer delay as a result of the additional hop. After path repair and link reconnection, the original path is only reselected if there is a fail in the selected path. To show video connectivity performance, the two hosts are configured respectively as video server and client with Videolan,, S transmitting via http to IP address of H. Cable is extracted and path reconfiguration is verified visually through video reception, very shortly interrupted.

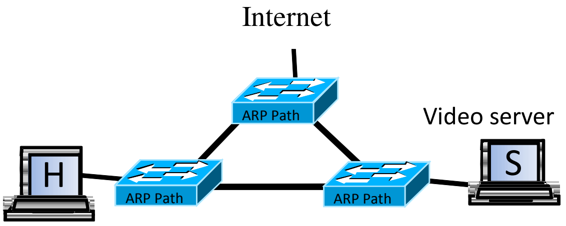


Figure 3. ARP Path Switches 100 Mbit/s Linux local demo scenario

Figure 4. ARP-Path bridges Linux local demo.

Most of the reconfiguration time (around 1070 msec with Soekris boards at 500 Mhz speed, half time with more powerful processors) is consumed by the link failure and ARP Request processing, that is performed in user space, instead of in Linux kernel. When the path is already established, ping round trip time is 900 microseconds.

2) 1 Gbit/s Openflow/NetFPGA demo (remote or video)

For high performance and ease of modification, an implementation based in Openflow [8] controlling NetFPGA boards [8] has been performed. A pure NetFPGA implementation is foreseen.The 1 Gbit/s demo network, shown at figs. 5 and 6, is located at Universidad de Alcalá and can be controlled remotely via VNC. Alternatively, a recorded video of the demo can be shown. Network consists of a fully connected mesh of 4 NetFPGAs (one per PC) acting as modified OpenFlow switches, with one central OpenFlow controller that implements ARP-Path logic for every switch.

Connectivity test and path reconfiguration after link failure is performed as for Linux demo with repetitive pings. Hosts are configured as video servers and also connect as clients and to video webs. All network links are connected and active at the same time and shortest (direct in this case) paths are selected. Paths are monitored by displaying the log of the NOX controller. Path set up time measured with repetitive pings between source and destination is 40-50 ms (if the MAC address is not in the table at the ARP-Path switches). and path reconfiguration unplugging takes 40-50 ms after link failure detection. Once the path is established, it takes only 0.5 msec for a ping round-trip time. ARP stress tests show handling of ARP Request rates of up to 1000 ARPs per second, using the *arp-sk* tool.

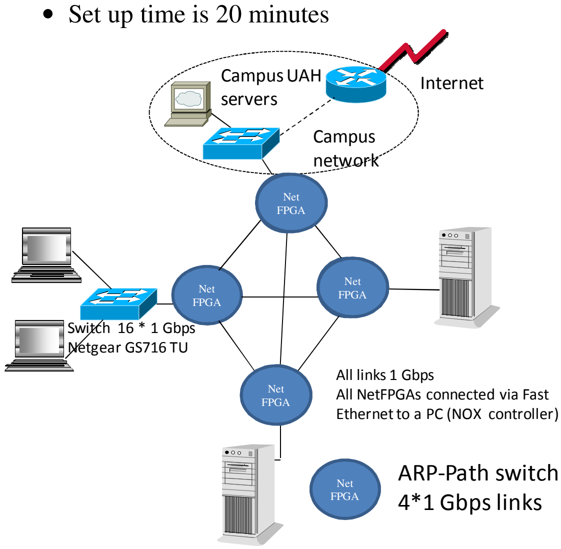
3) Equipment at conference venue

The required equipment (provided by us) is listed below:

- Two laptops and AC/DC power supplies
- VNC client equipped in laptops
- AC power for laptops

4) Demo requirements

- Internet access (preferably wired)
- AC power
- Space required: table 60 cm x 80 cm
- Set up time is 20 minutes



.

Figure 5. Remote demo schema of Openflow/NetFPGA 1 Gbit/s

Figure 6. 1 Gbit/s NetFPGAs demo set up at UAH lab

C. Acknowledgment

This work was supported in part by grants from Comunidad de Madrid and Comunidad de Castilla la Mancha through Projects MEDIANET-CM (S-2009/TIC-1468) and EMARECE (PII1I09-0204-4319).

D. References

[1] IEEE 802.1D-2004 IEEE standard for local and metropolitan area networks-Media access control (MAC) Bridges. Available online: standards. ieee.org/getieee802 /802.1.html.

[2] M. Seaman. Shortest Path Bridging. Available online: ieee802.org/ 1/files/public/docs2005/new-seaman-shortestpath-par-0405-02.htm.

[3] Transparent interconnection of lots of links (TRILL) WG. Available on line at: ietf.org/html. charters/trill-charter.html

[4] Ibanez G. et al. ARP-Path Ethernet Switching: On-demand Efficient Transparent Bridges for Data Center and Campus Networks. LANMAN May 2010. Available on line at: hdl.handle.net/10017/6298

[5] Elmeleegy, Khaled and Cox, Alan L. EtherProxy: Scaling The Ethernet By Suppressing Broadcast Traffic. Proceedings of IEEE INFOCOM 2009, Rio de Janeiro, Brazil.

[6] Omnet Simulatior. Available on line: omnetpp.org

[7] NRS Reference Networks. Available on line at: ibcn.intec.ugent.be/ INTERNAL/NRS/index.html

[8] NetFPGA: netfpga.org

[9] Openflow: openflowswitch.org

[10] Ebtables.http://ebtables.sourceforge.net