Dynamic Load Routing/Path Diversity in a Network of ARP-Path NetFPGA Switches

Elisa Rojas, Jad Naous, Guillermo Ibáñez, Carlos Recio

***Abstract*— We show the native dynamic load routing of ARP path protocol at level of per-host path diversity with a network of four independent ARP path switches. Each ARP path switch is made of a 4*1 Gbps NetFPGA board and a 5501 Soekris board running CentOS. We also compare network throughput versus spanning tree protocol, showing full network links usage with ARP path. Multiple video streams are set up among multiple virtual machines to show the diversity of paths selected. Network reconfiguration speed is verified by unplugging links and observing video transitions. Connection of the whole network to Internet is also shown to demonstrate transparency and compatibility.**

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

I. Introduction

Current standards like Shortest Path Bridges (SPB) [1] and Routing Bridges (TRILL) [2] are proposals to implement shortest path bridging in switched networks to overcome current limitations of the spanning tree protocol [3]. They use a link state routing protocol in layer two to obtain shortest paths between bridges. This means increased complexity to compute shortest paths, additional mechanisms to prevent loops and to balance the load. ARP-Path protocol takes a different approach. It is a pure bridging protocol that sets up an on-demand path between hosts when needed, by snooping ARP Request packets. This makes ARP path to natively distribute the load dynamically and to select low latency paths. We demonstrate this features with a compact, line speed implementation on a 4*1 Gbps NetFPGA board controlled by a Soekris board for management and configuration purposes, as shown in figs. 1 and 2.

II. ARP-Path Protocol

We summarize here the basic operation of the ARP path protocol. [4]

A. ARP-Path Path setup

The ARP-Path protocol relies on the race between flooded ARP requests to establish the fastest path. Note that only ARP frames (or special broadcast frames in failure cases) discover or create new paths.

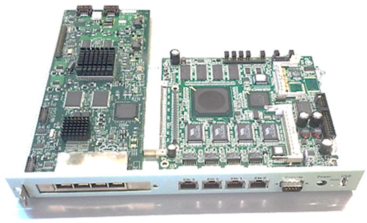

Figure 1. ARP Path line speed switch (NetFPGA and Soekris 5501 board)

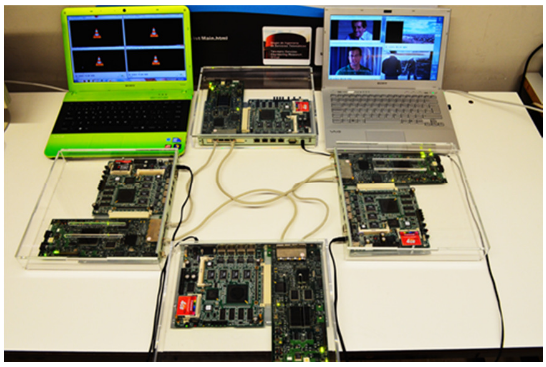

Figure 2: Demo network with ARP-Path 4*1 Gbps switches

1) ARP-Path Broadcast Path Discovery (ARP Request)

When host S wants to send an IP packet over Ethernet to host D over IP, it needs D's MAC address. If the mapping of D's IP address to D's MAC address is not in S's ARP cache, S broadcasts an ARP Request, B, for D's MAC address (Figure 2-a). Ingress bridge 2 receives the frame from S and temporarily associates (locks) S's MAC address to the ingress port. Unlike traditional learning switches, further broadcast frames from S arriving to other input ports of bridge 2 will be discarded because they arrived over slower paths. S's address is now in a locked state and bridge 2 broadcasts B on all other ports (Figure 3-b). 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's MAC address, they are discarded (Figure 3-c). 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 reverse path from D to S (Figure 3-c).

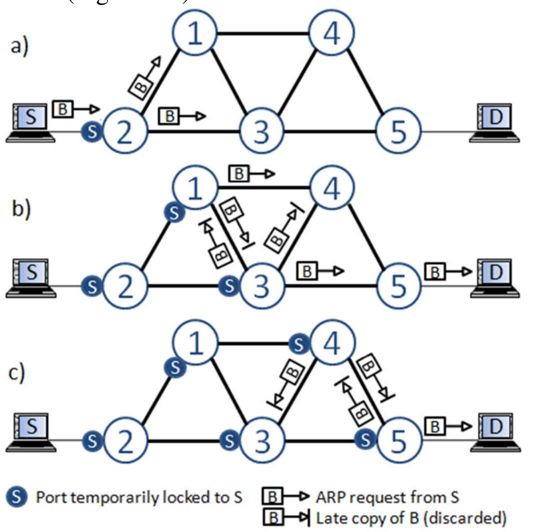

Figure 3: ARP-Path path set up from host S to host D. The small bubbles on the links show which bridge port is temporarily associated (locked) to S's address

2) ARP-Path Unicast Discovery (ARP Reply)

The next step is 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, with S's MAC address as destination address. Given the temporary reverse path back to S that was established by the ARP Request frame, U can be delivered with no further broadcasts. Like the ARP Request frame, U establishes a path from S to D for other unicast packets from S to D.

3) Path Recovery (Network reconfiguration)

When a unicast frame arrives at a bridge, the bridge may not know the output port for the frame's destination MAC address. The entry could have expired, or a link or a bridge might have failed. The unknown unicast frame is then *looped back* towards its source edge bridge using the *reverse forwarding* mode for unknown unicast frames. When the frame frame reaches the source edge bridge, this issues a new ARP Request packet to find a new path. Only paths actually used are recovered and only when needed. There is no need to flush all learnt MACs across the bridged network, the new ARP Request will create the new path needed. Only the paths traversing reconfigured links are recovered.

III. ARP Path Network Demonstration

A. Path diversity

First part of the demo shows path diversity among flows from different hosts. It consists of four video flows that are set up from four virtual machines (see fig. 4) of the same laptop to another four virtual machines at another laptop. Sequentially, each virtual host sets a path with a different virtual host at the other side of the network. The selection of different paths is observed by the LEDs at NetFPGA boards and also by inspection of switches forwarding tables. Once all paths are set, links are sequentially unplugged: video flows using the unplugged link freeze shortly for a fraction of a second and resume while flows using alternative paths are not affected. Plugging again the links and then unplugging a link of the active path will force all video flows to repair their path in a short time.

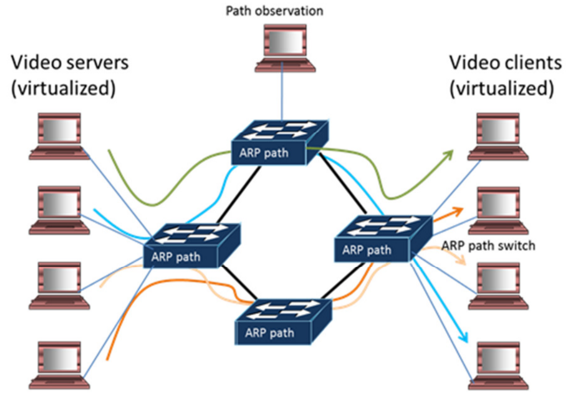

Figure 4: Path diversity/dynamic load routing demo

B. Maximum throughput vs. spanning tree

Demo set up for throughput comparison of ARP path with spanning tree is shown in fig.5. Using *iperf*, we send UDP or TCP flows from upper/lower host in the right to the respective upper/lower host in the left.

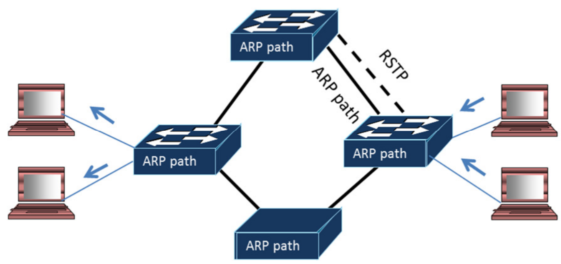

Figure 5: Network throughput comparison: ARP path versus spanning tree.

Fig. 6 shows throughput in Mbps at each receiver host for ARP path protocol with TCP and UDP and fig. 7 for STP. For STP the NetFPGA project *reference_switch* is loaded in NetFPGA boards. Although nominal limit would is 1 Gb/s per flow, the hosts limit the maximum throughput to 950 and 820 Mbps respectively. With ARP path these maximum values are achieved without significant packet loss (0,12%). Flows go through parallel paths in the network and do not compete for bandwidth.

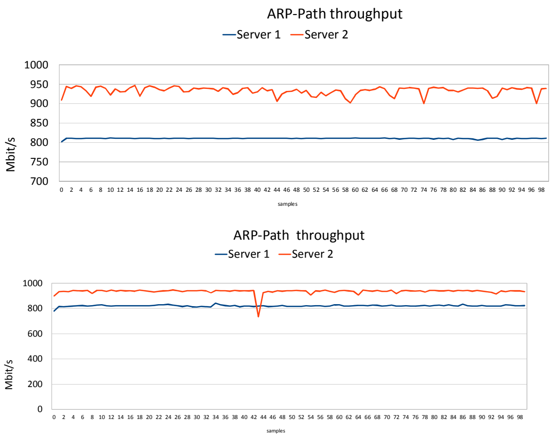

Figure 6: Throughput per host link ARP path protocol (UDP and TCP traffic respectively)

Figure 7: Throughput per host link with spanning tree protocol (UDP and TCP traffic respectively).

With spanning tree, available single link (1 Gbps) bandwidth is clearly shared among the two server links. Flows compete for the available bandwidth. Due to the high bandwidth restriction, measured packet losses are on average 44% at each host for STP.

C. *Internet access*

By connecting to Internet by sharing the internet connection of one of the hosts, as shown in fig. 8. All hosts can get full internet access and show full compatibility with DHCP and other services. Any web internet service can be tested: Web, FTP, Skype, video streaming.

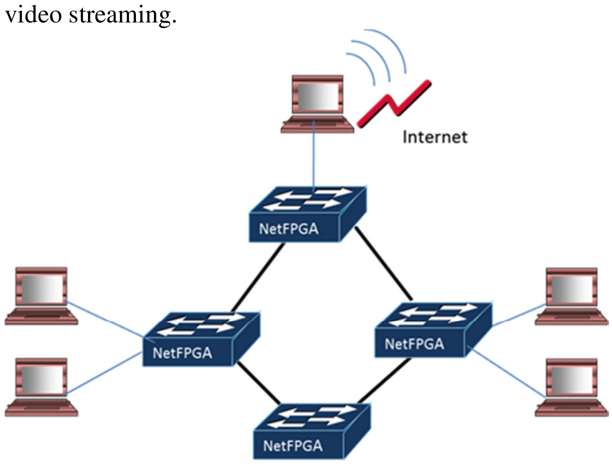

Figure 8: Internet connection via a shared connection.

D. *Equipment at conference venue*

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

- Two/four laptops and AC/DC power supplies
- AC power for laptops

Demo requirements are: Internet access (wireless or wired), AC power. Space required: table 60 cm x 120 cm, set up time is 40 minutes.

IV. Acknowledgment

This work was supported in part by grants from Comunidad de Madrid through Project MEDIANET-CM (S-2009/TIC-1468) and Universidad de Alcalá through project CMACs (UAH2011/EXP-016).

References

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

[2] Allan, D., Ashwood-Smith, P., Bragg, N., Farkas, J., Fedyk, D., Ouellete, M., Seaman, M., and Unbehagen, P. Shortest path bridging: Efficient control of large Ethernet networks. Communications Magazine, IEEE, 48(10):128-135, October 2010.

[3] Transparent interconnection of lots of links (TRILL) WG. https://datatracker.ietf.org/wg/trill/charter/

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

[5] NetFPGA: http://www.netfpga.org