ARP-Path bridges only accept frames from a particular source at the port that receives the first multicast or broadcast frame from that source. Unlike ARP Requests, other multicast and broadcast frames do not establish new paths.

2.1.4 Path Repair

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 Path Repair protocol emulates an ARP exchange to establish a new path, using PathFail, PathRequest, and PathReply messages. PathRequest messages are similar to ARP Request frames and establish the new path to the unknown destination. Thus a full path from to the destination end-host is restored.

2.2 Advantages

The protocol has several important advantages over other protocols that build routes *a priori* before any packet transmissions.

- *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 hosts and bridges.
- *Simplicity*: Bridges mainly behave as learning switches with optional ARP proxying to reduce broadcasts.
- *Load distribution and path diversity*
- *Scalability*: ARP broadcast traffic can be reduced dramatically by implementing ARP Proxy function inside the switches as shown in [5].

3. ARP-PATH OVER NETFPGA DEMO

The objective of the NetFPGA [6] implementation is to understand the robustness and throughput of ARP-Path transparent bridges in 1 Gbit/s wired networks, without the spanning tree protocol or any ancillary routing protocol operating at layers two or three.

To test the implementation we will use a PC with four NetFPGAs installed, that will behave as separated switches, connected between them and to external hosts. A user interface was created to visualize results that will launch the scripts that run the demonstration, and will build graphs to show the latencies obtained.

3.1 ARP-Path vs. STP

The main goal is to compare ARP-Path's behaviour with that of STP. The setup will consist of one desktop with 4 NetFPGAs and 2 NICs. The connections are to be physically as indicated in Figure 2.

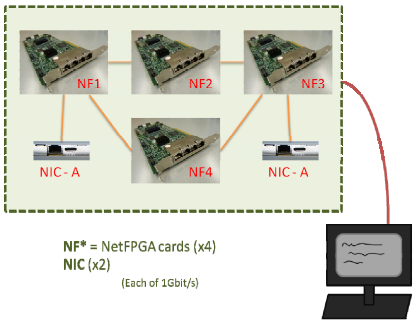

Figure 2: 4 NetFPGAs and 2 NICs connection, for the test.

The four NetFPGAs (NF1, NF2, NF3 and NF4) operate as ARP-Path bridges in one part of the demo and as STP bridges (NICs operating as separate STP bridges managed using Linux's bridge_utils) in another. We will show that ARP-Path chooses lower latency paths as opposed to STP that builds a routing tree rooted at an arbitrary switch.

3.2 ARP-Path switches Path repair

The second goal is to show that the protocol path repair is fast. Hosts *A* and *B* will start a video streaming communication. Host *A* will act as a HTTP server and *B* will connect to it and start streaming a video. We show ARP-Path's Path Repair's effectiveness after sucessive link failures and its minimal effect on the streamed video. See Figure 3.

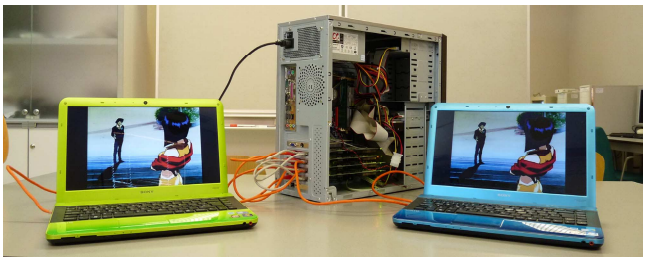

Figure 3: Two hosts connected to a PC with four NetFPGA installed

4. ACKNOWLEDGMENTS

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).

5. 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] Ibañez, G., Carral, J. A., Martinez, A. G., Arco J. M., Rivera, D. and Azrorra, A. Fastpath 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, K. and Cox, A. EtherProxy: Scaling the Ethernet by suppressing broadcast traffic. *Proceedings of IEEE INFOCOM,* 2009, Rio de Janeiro, Brazil.

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