Torii HLMAC: Distributed, Fault-tolerant, Zero Configuration Data Center Architecture with Multiple Tree-based Addressing and Forwarding

Elisa Rojas, Guillermo Ibañez
elisa.rojas@uah.es, guillermo.ibanez@uah.es
University of Alcala. Campus Externo, Alcalá de Henares 28805 Spain

ABSTRACT

This paper describes Torii-HLMAC (鳥居HLMAC), a scalable, fault-tolerant, zero-configuration data center network fabric architecture (currently under final evaluation) as a full distributed alternative to Portland for similar multiple tree (fat tree) network topologies. It uses multiple, fixed, tree-based positional MAC addresses, used for multiple path table-free forwarding. Addresses are assigned by simple extension of the Rapid Spanning Tree Protocol. Torii-HLMAC enhances the Portland protocol advantages of scalability, zero configuration and high performance and adds instant path recovery, distributed address assignment. ARP broadcast may use ARP Proxy.

Categories and Subject Descriptors

C.2.5. [**Computer-Communication Networks**]: Local and Wide-Area Networks – *Ethernet*.

General Terms

Design, Experimentation,Verification.

Keywords

Ethernet, Tree-based routing, Routing bridges, Data Centers, Fat Trees, Shortest Path Bridges, Spanning Tree.

1. INTRODUCTION

Data center networks are increasingly relying on Ethernet and flat layer two networks due to its excellent price and performance ratio and configuration convenience. The replacement, by economic reasons, of the scale up model by the scale out model [1], using a high number of commodity servers and switches is driving data center networks to high scale dimensions. Different approaches to implement a data center fabric have been recently proposed to overcome the limitations of Spanning Tree protocol (ST) and the configuration complexity of Multiple Spanning Tree Protocol. Portland [2] is a recent protocol proposal for data centers that uses centralized control, location based pseudo MAC addresses and Up/Down turn prohibition to prevent loops. Addresses are assigned to hosts and switches by a discovery protocol.

ACM CoNEXT Student Workshop, December 6, 2011, Tokyo, Japan.

In this ongoing work we explore a combination of distributed functions to make forwarding in fat trees simple and more scalable. Torii-HLMAC architecture aims to improve Portland with alternative, simpler and distributed mechanisms. It uses topological pseudo MAC addresses, but multiple simple addresses (inspired by TRE [3]) to facilitate multipath forwarding as well as fault tolerance, direct frame routing without tables and on the fly alternative path selection after link failure.

2. PROTOCOL DESCRIPTION

A. Tree-based Multiple Addresses structure and automatic assignment with Extended RSTP

Torii-HLMAC requires each bridge to be assigned a Hierarchical Local MAC (HLMAC) address for every port connected upstream as shown in Fig. 1. HLMAC addresses are local MAC addresses (U/L bit is set to 1). The 46 bits available for addressing purposes (after removing the local or global bit and the multicast bit) encode by default up to 6 different hierarchical levels, with 6 bits for the first and 8 bits for each other level. The HLMAC of a bridge is expressed in the dotted form a.b.c... as the chain of designated port IDs a, b, c, ... traversed in the descending path from the Root Bridge to the bridge to which the address is assigned.

To build the spanning tree and assign hierarchical addresses to the bridges, a modified version of RSTP is used, which is defined in HURP [4]. Once the root bridge is set, which gets 0.0.0.0.0.0 as the HLMAC, the process of building the spanning tree from the root to the leaves starts. This iterative process consists of BPDUs being sent by the parent bridge including the number of the Designated Port. These numbers are 1,2,3,4, which correspond to the pod number that the port is connected to, as shown in Fig 1. For instance, the first aggregation switch at pod 1 has 1.1. (from core switch 1 and designated port 1) and 2.1. (from core switch 2 and designated port 1) as HLMAC addresses.

As a result, each node gets one or more (up to four) topological tree addresses (HLMAC). Since there are four core roots, there will be four alternative HLMAC addresses at the edge switches and the prefix will be used to distribute traffic on a hash base.

Torii is scalable for bigger topologies and could also be used in fat trees (with “fatter” links towards the core).

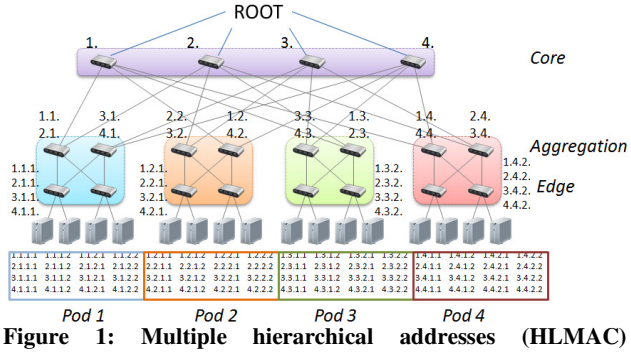

Figure 1: Multiple hierarchical addresses (HLMAC) assignment for Torii with extended Rapid Spanning Tree Protocol from virtual Root node.

B. Tree-based Forwarding

Routing of every frame is directly performed via address decoding. Once the HLMACs are set, Torii switches need to distinguish among broadcast/multicast and unicast frames, and identify the direction of the frame: "going up" or "going down" (this is done thanks to the frame input port). Once those two parameters are known, the logic applied in each switch of the topology is the following:

If frame is **BROADCAST** or **MULTICAST**: (see Fig. 2)
- If frame goes UP:
  - If switch is edge → host MAC to HLMAC (prefix chosen by a hash)
  - Forward frame through the HLMAC port *
  - Down broadcast frame ***
- Else if frame goes DOWN:
  - If switch is edge → HLMAC to host MAC
  - Down broadcast frame ***

Else if frame is **UNICAST**: (see Fig. 3)
- If frame goes UP:
  - If switch is edge → host MAC to HLMAC (prefix chosen by a hash)
  - Forward frame through the HLMAC port **
- Else if frame goes DOWN:
  - If switch is edge → HLMAC to host MAC
  - Forward frame through the HLMAC port *

*: Forwards through the next port according to the HLMAC address (up port if the frame comes from a down one or viceversa)

**: Same as * but, in unicast, sometimes frames do not need to reach the core switch if there is a shorter path, in this case frame is not forwarded to the core (indicated by the HLMAC prefix).

***: Broadcast only through the ports located down in the hierarchy except through the input port.

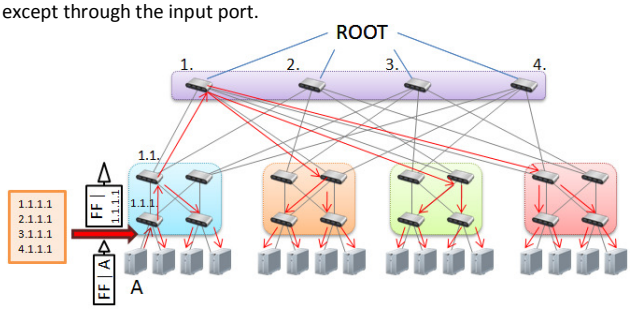

Figure 2: Broadcast frame from host A. The broadcast address remains the same while the A address is translated into 1.1.1.1 when prefix 1 has been chosen at the edge by hash.

As it can be seen multicast and broadcast forwarding are performed across the spanning tree as it occurs in classical Ethernet, while unicast forwarding is quite similar but frames go right to the destination and sometimes they can take a shortcut. ARP Proxies may be used at edge bridges.

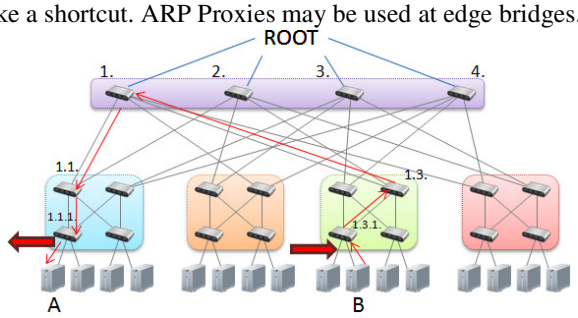

Figure 3: Unicast frame from A to B. Both address (A and B) are translated at the edge switches, which already know them from the previous ARP messages. The prefix (core switch) is chosen by a hash of both addresses so that the communication is bidirectional. In this case A goes 1.1.1.1 and B goes 1.3.1.2.

3. EVALUATION

Torii HLMAC has been simulated in Omnet. The implementation, coded in C++, relies on the MACRelayUnit module (from inet/linklayer/etherswitch). The base has been modified so that it acts as a Torii switch. STP is not implemented, however, the STP BPDUs are given as parameters in the simulation instead.

4. CONCLUSIONS

Torii-HLMAC improves Portland in several ways: multiple addresses are automatically assigned in a distributed form without duplicates, instead of by a centralized module. Routing is completely distributed, and performed solely based on the destination tree-based HLMAC address used, without routing tables at bridges, allowing high speed forwarding. In case of a link failure in a path, the bridge instantly selects an alternative path to reach the destination host and also notifies both edge switches serving origin and destination so that the non valid path is not chosen again, for a while. The topology scales up to 6 levels plus roots and more, if needed.

5. ACKNOWLEDGMENTS

This work is 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).

6. REFERENCES

[1] A. Vahdat et al. Scale Out Networking in the Data Center. IEEE Micro, July/August 2010

[2] R. Mysore et al. PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric. In ACM SIGCOMM, August 2009.

[3] G. Ibáñez et al. Evaluation of Tree-based routing Ethernet. IEEE Communication Letters, IEEE June 2009 Vol. 13 No 6 pp. 444 – 446. DOI: 10.1109/ LCOMM.2009.090469

[4] G. Ibáñez et al. HURP/HURBA: Zero-configuration hierarchical Up/Down routing and bridging architecture for Ethernet backbones and campus networks. Computer Networks. Vol. 54, Issue 1, 15 January 2010, pp 41-56. http://dx.doi.org/10.1016/j.comnet.2009.08.007