Ethernet Technology Summit

Torii-HLMAC: Torii-HLMAC: Fat Tree Data Center Architecture

Elisa Rojas
elisa.rojas@uah.es
University of Alcala
(Spain)

Ethernet Technology Summit

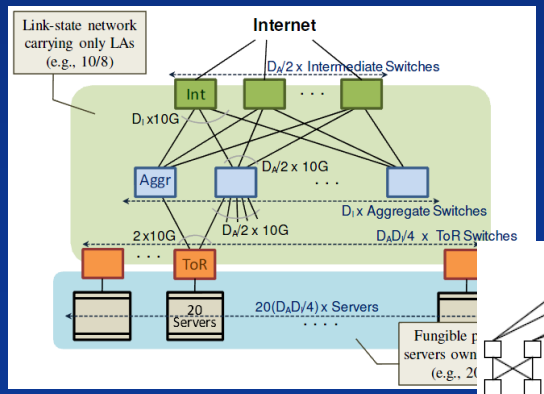
Outline

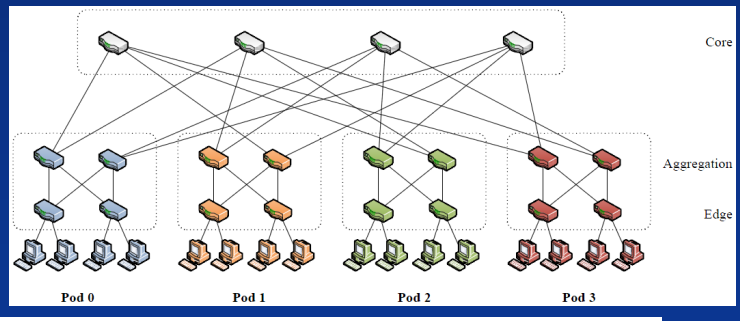
- Introduction
- Protocol description
  - *Tree-based Multiple Addresses structure and automatic assignment with Extended RSTP*
  - *Tree-based forwarding*
  - *Tree-based path repair*
- *Evaluation*
  - *Simulation of Torii-HLMAC*
  - *Other issues:*
    - *Use of Virtual Machines at hosts, HLMAC Address Assignment Alternatives, Inter-L2 Mobility, Generalization to any data center topology*
- *Conclusions*

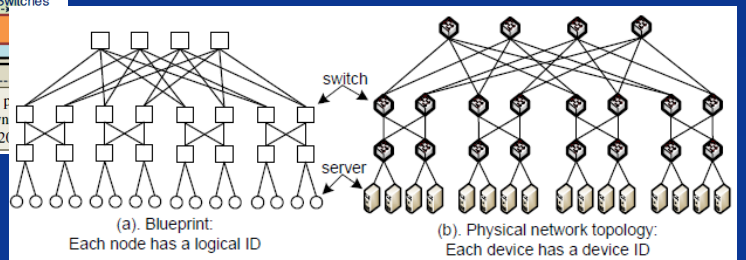
Introduction

- **Data center** networks are increasingly relying on **Ethernet** and flat layer two networks
  - Due to its excellent price, performance ratio and configuration convenience
- **Scale-out** model over **scale-up** model

  → High scale dimensions → Limitations of RSTP
- Recent architecture proposals:
  - **VL2**
  - **PortLand**
  - **DAC**
    - Blueprint

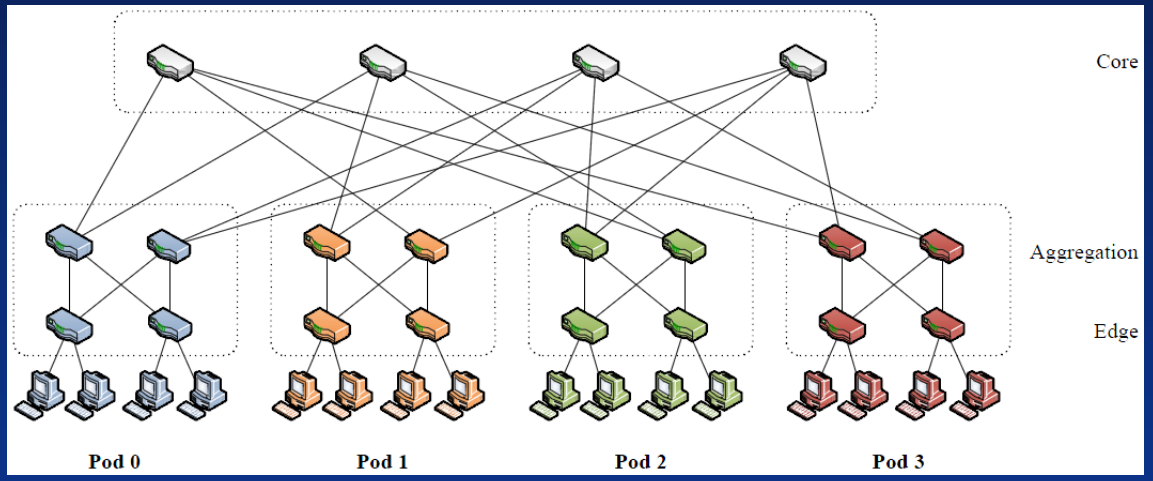






Introduction

- So… if we have the advantages of using this type of **topology**…

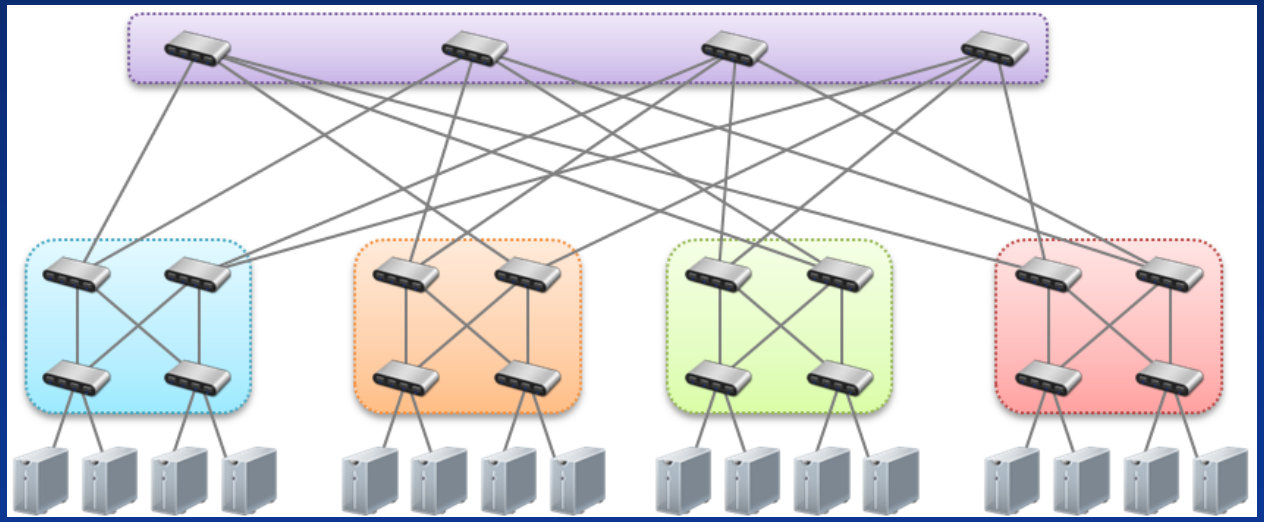


…why not make the most of it and consider it as an **specific topology** to enhance the whole architecture and data center protocol?

➔ Torii-HLMAC

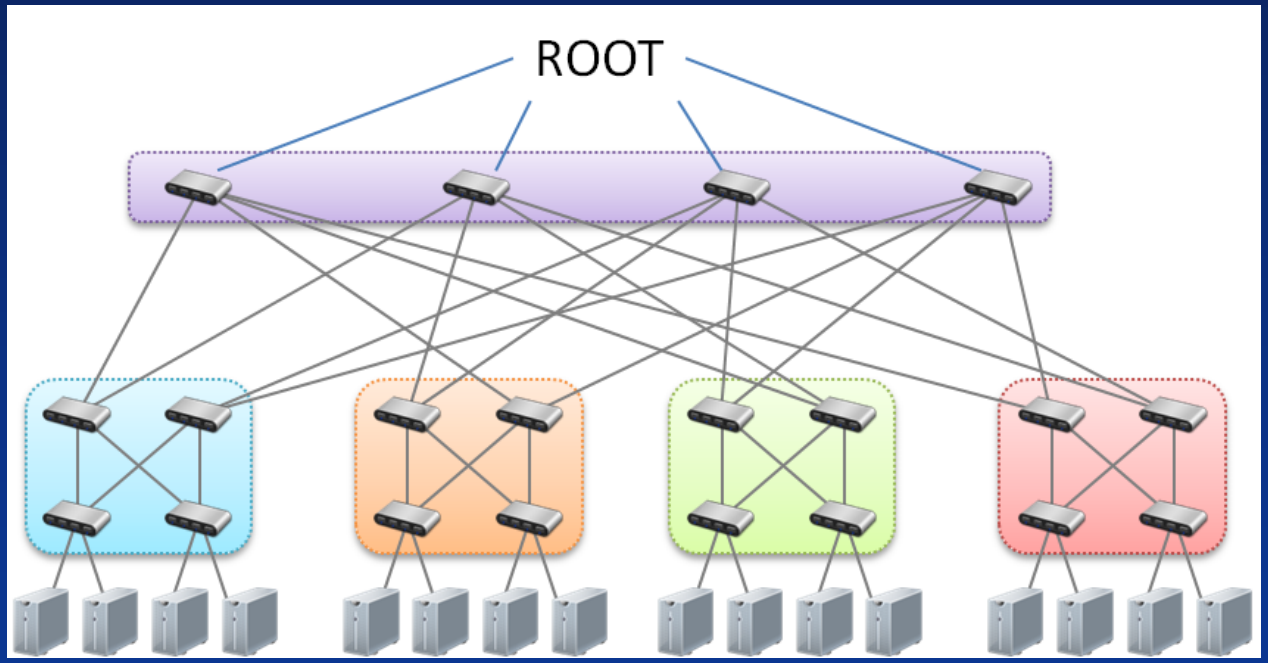
Protocol description

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



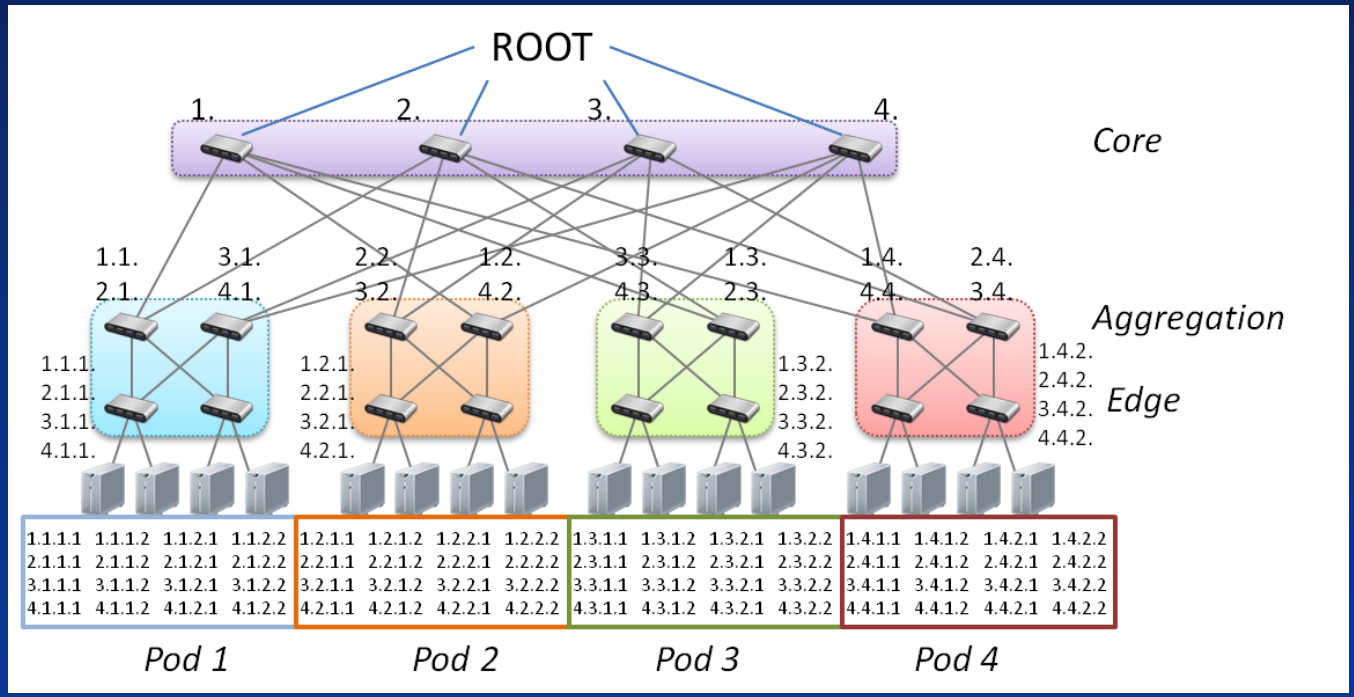
Protocol description

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



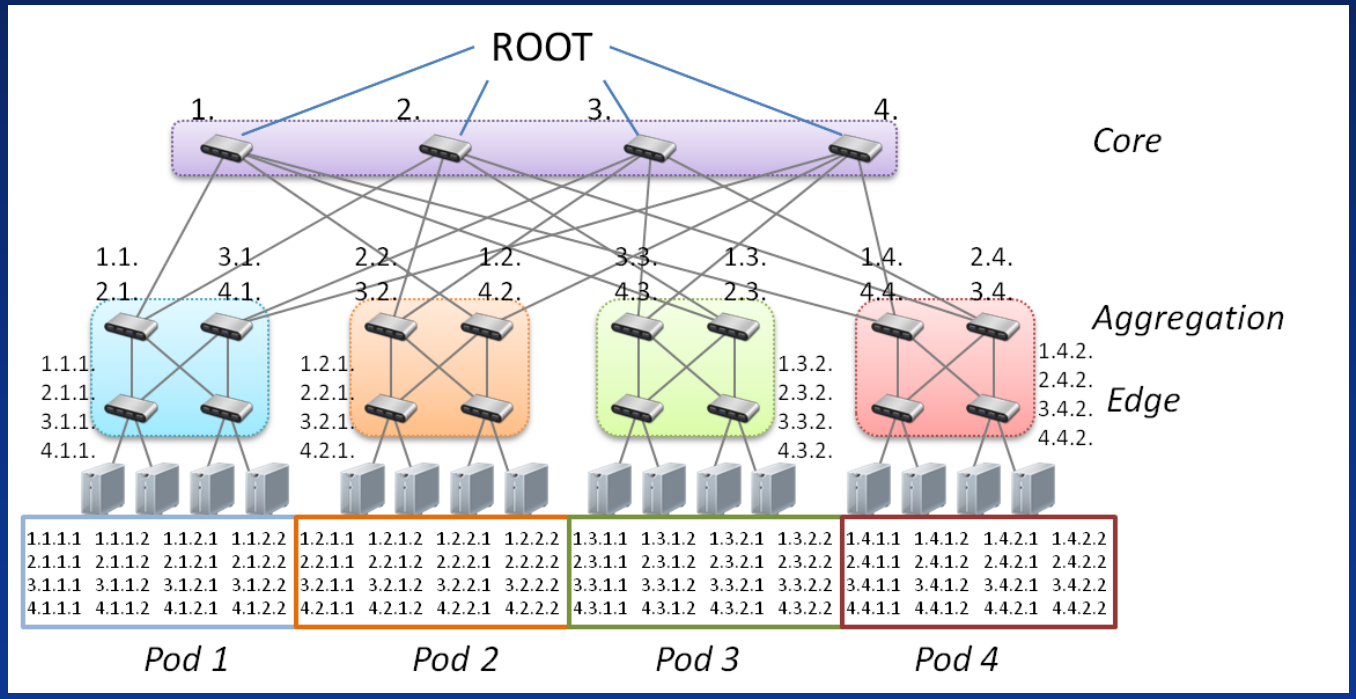
Protocol description

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



Protocol description

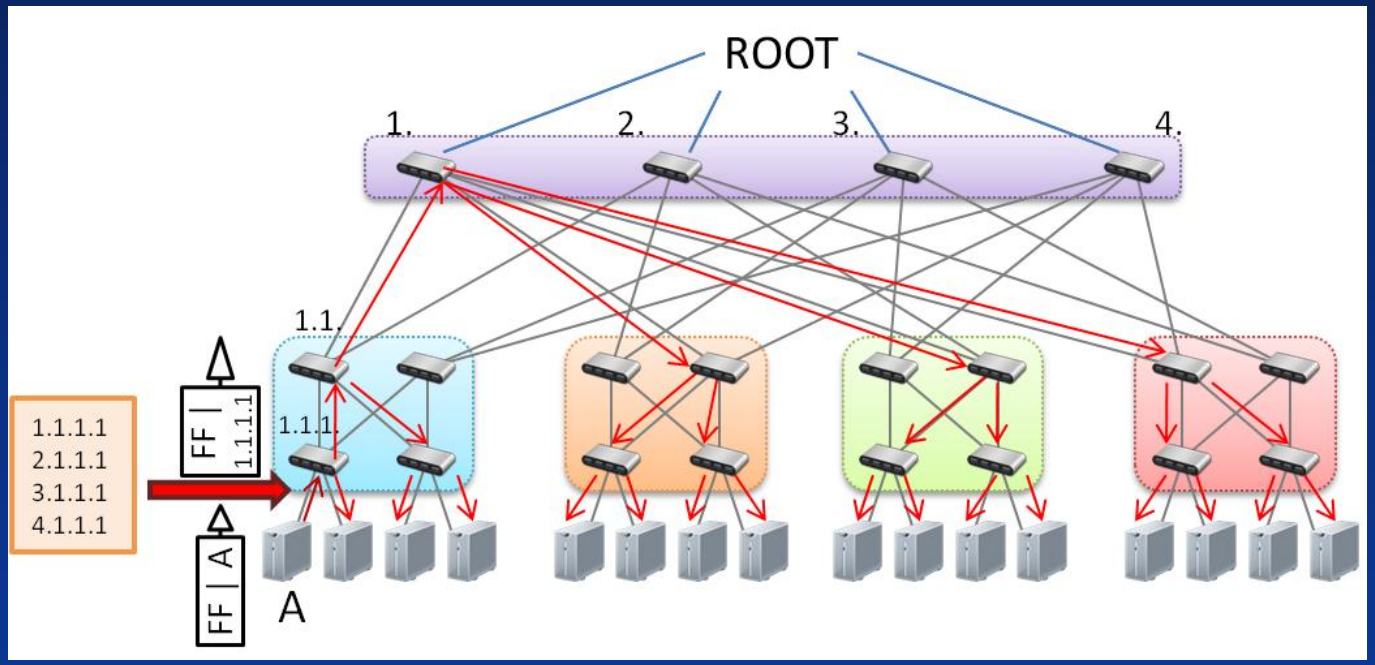
- *HLMAC are local MAC (U/L bit=1)*
  - *Almost 6 bytes (6bits+5x8bits) → ROOT is 0.0.0.0.0.0*



  - *Address 1.1.1.1 = 1.1.1.1.0.0, (in fact the first byte will not be 1, since the U/L bit will be set to 1, but it is omitted)*

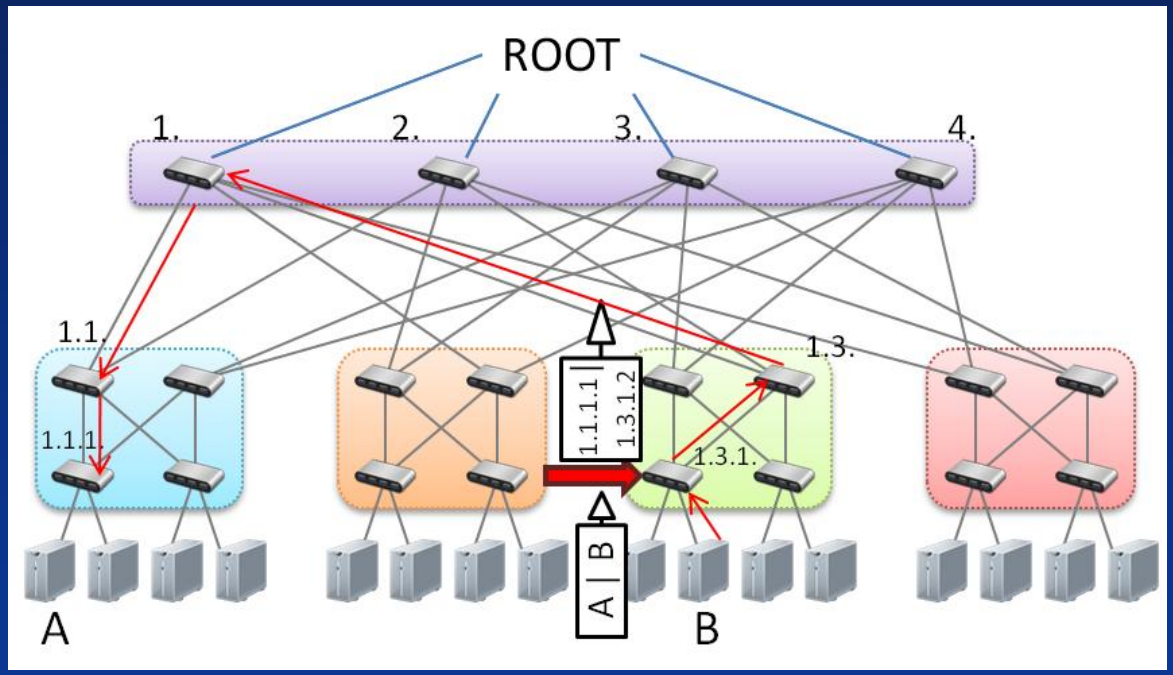
Protocol description

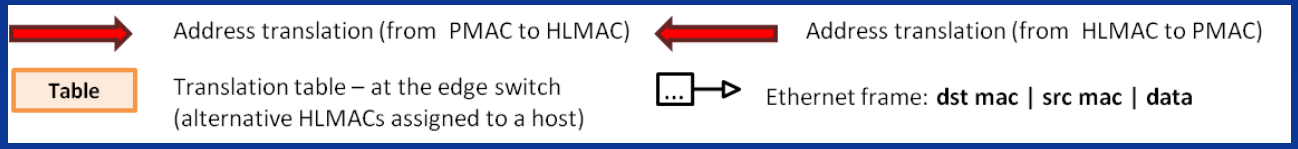
- *Tree-based forwarding*
  - *Broadcast and Multicast*



Protocol description

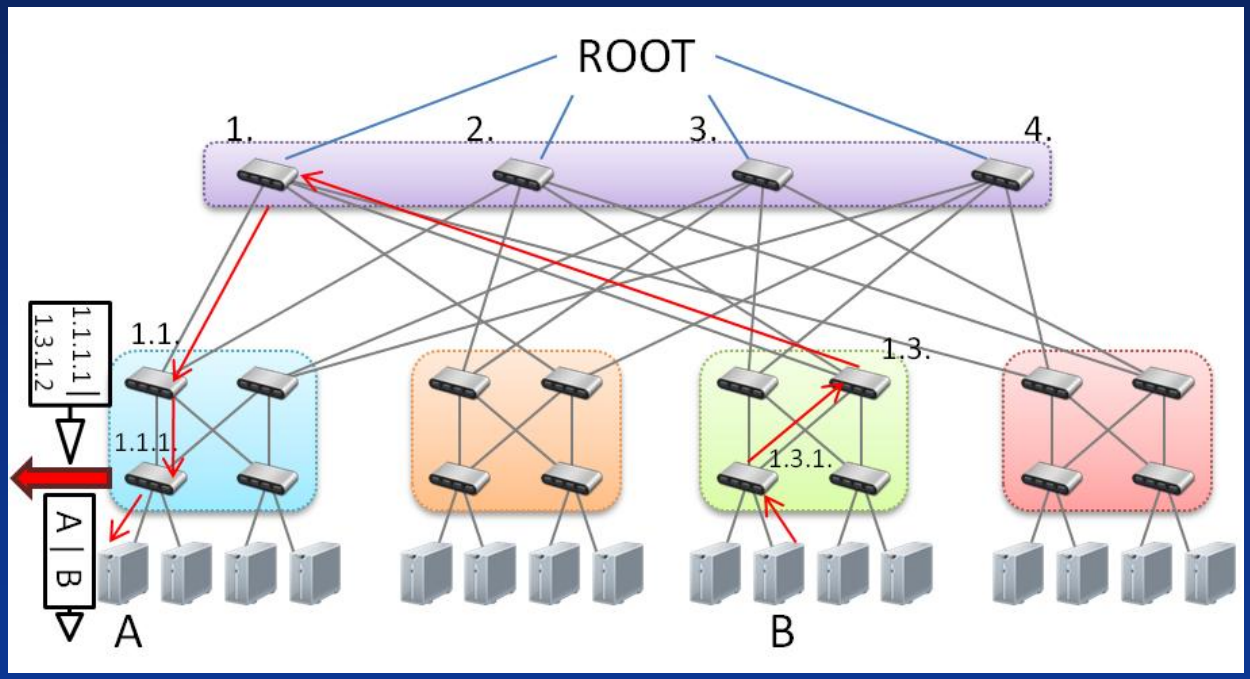
- *Tree-based forwarding*
  - *Unicast*

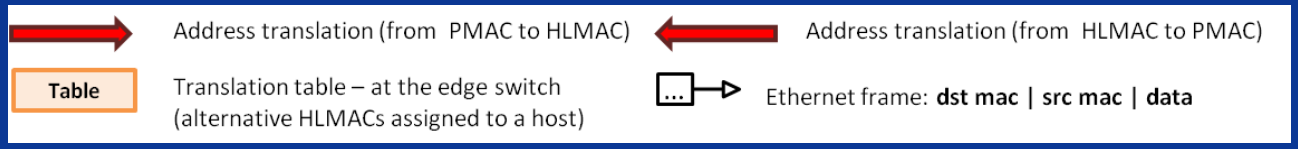




Protocol description

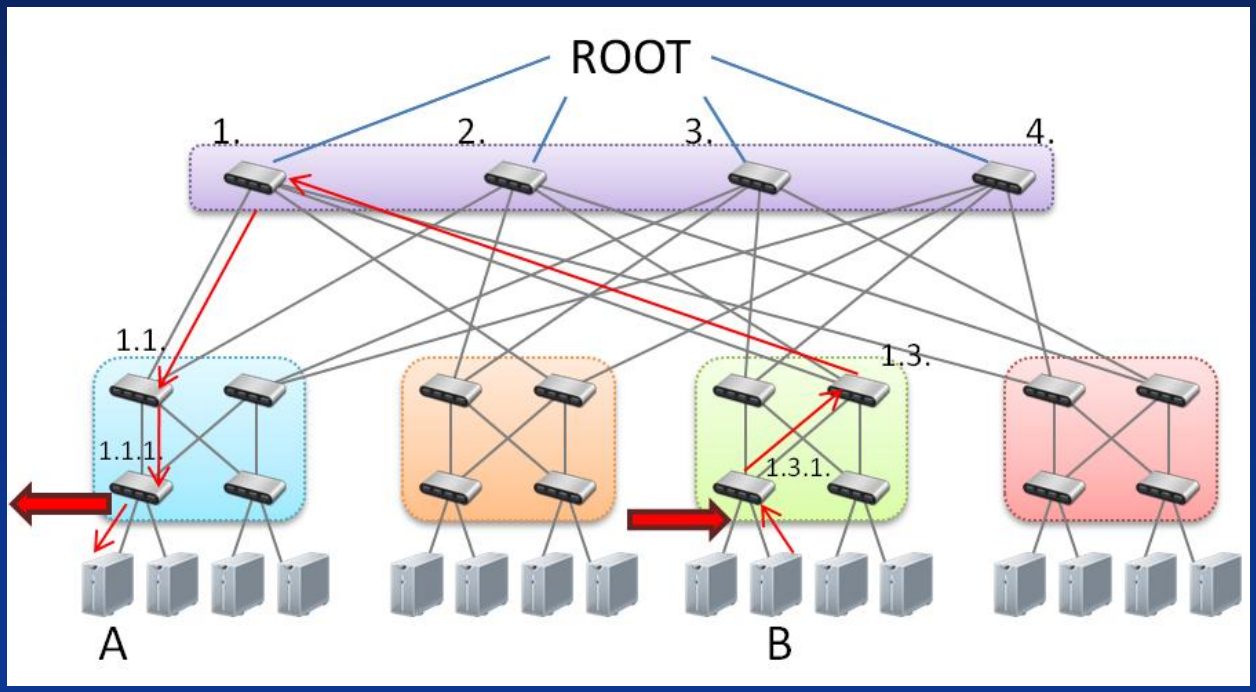
- *Tree-based forwarding*
  - *Unicast*





Protocol description

- *Tree-based forwarding*
  - *Unicast*



Address translation (from PMAC to HLMAC)

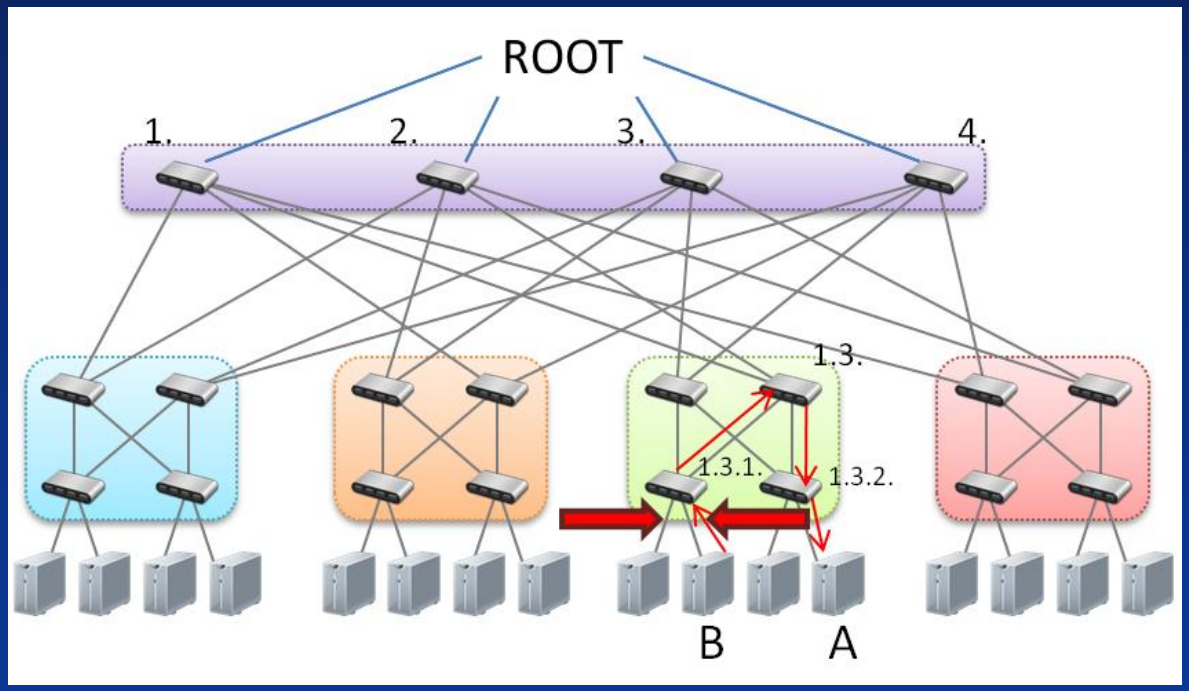
Address translation (from HLMAC to PMAC)

Table — Translation table – at the edge switch (alternative HLMACs assigned to a host)

Ethernet frame: **dst mac | src mac | data**

Protocol description

- *Tree-based forwarding*
  - *Unicast*



Address translation (from PMAC to HLMAC)

Address translation (from HLMAC to PMAC)

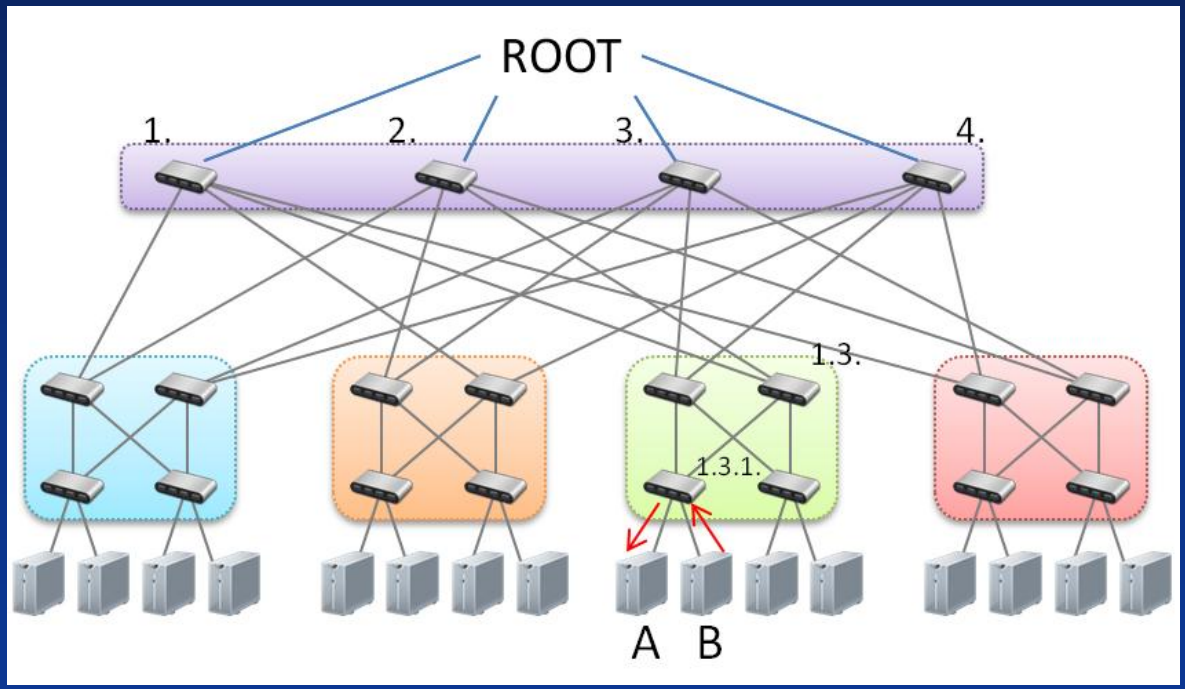
Table

Translation table – at the edge switch (alternative HLMACs assigned to a host)

Ethernet frame: **dst mac | src mac | data**

Protocol description

- *Tree-based forwarding*
  - *Unicast*



Address translation (from PMAC to HLMAC)

Address translation (from HLMAC to PMAC)

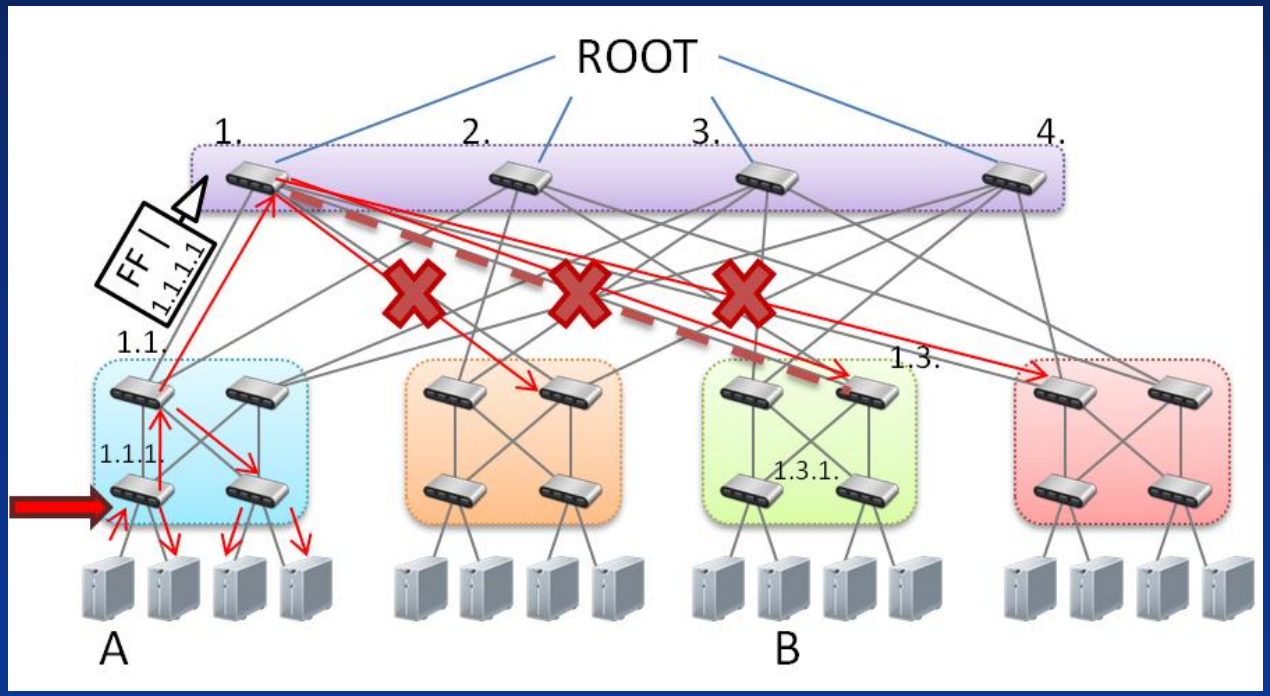
Table

Translation table – at the edge switch (alternative HLMACs assigned to a host)

Ethernet frame: **dst mac | src mac | data**

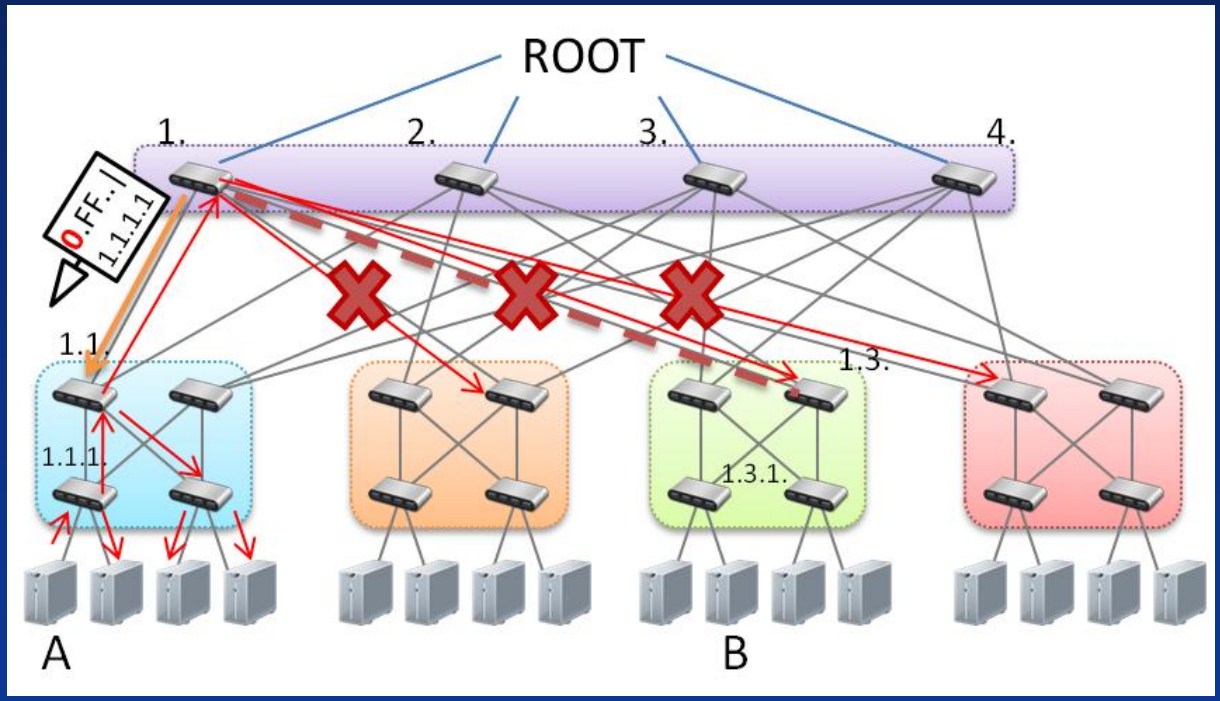
Protocol description

- *Tree-based path repair*
  - *Broadcast and Multicast*



Protocol description

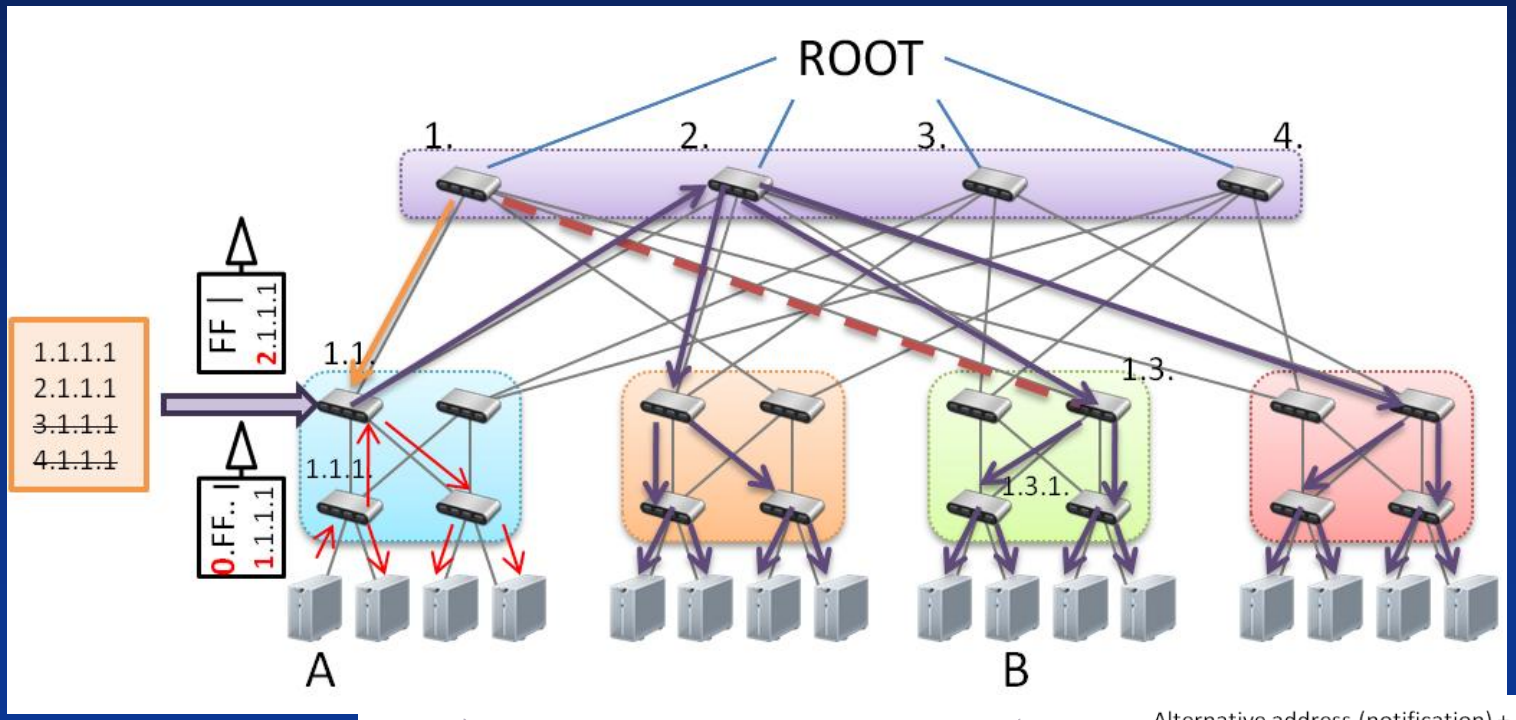
- *Tree-based path repair*
  - *Broadcast and Multicast*



  - *Path repair looks for the **first alternative** to avoid duplicates*

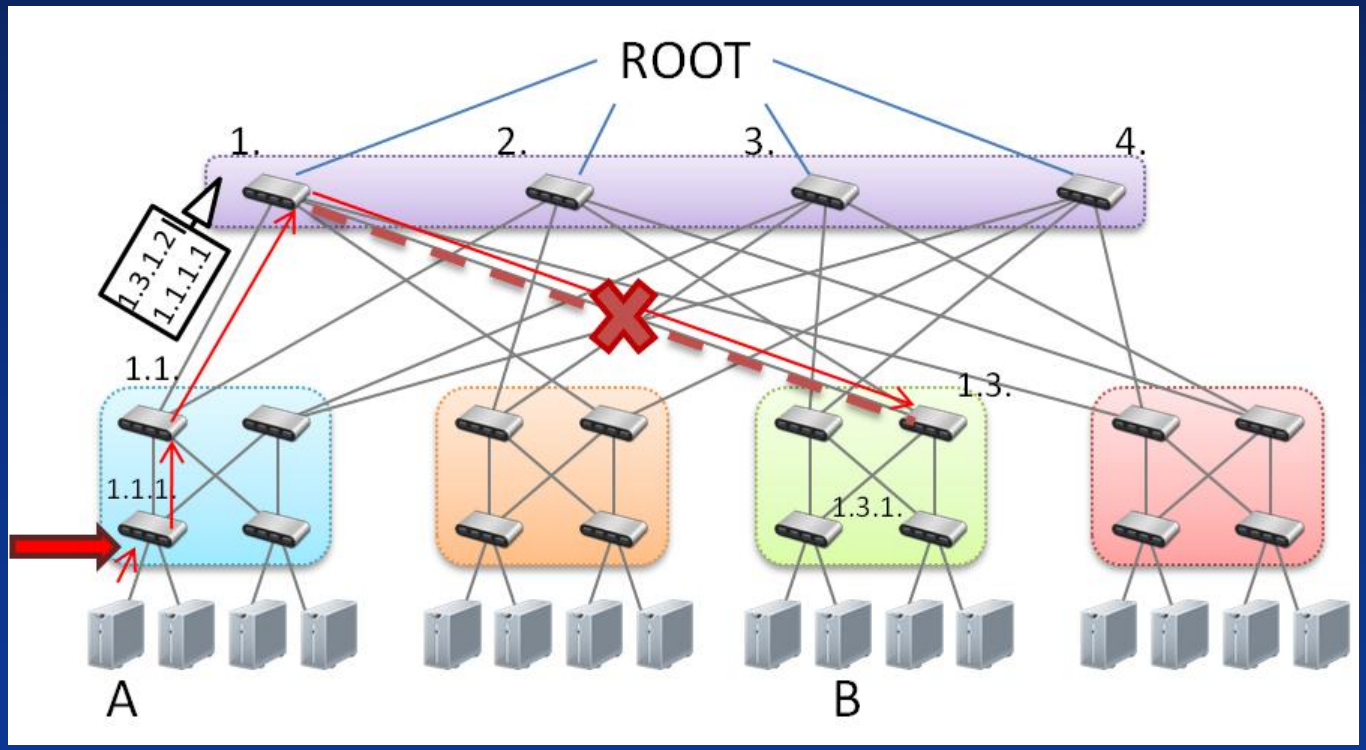
Protocol description

- *Tree-based path repair*
  - *Broadcast and Multicast*



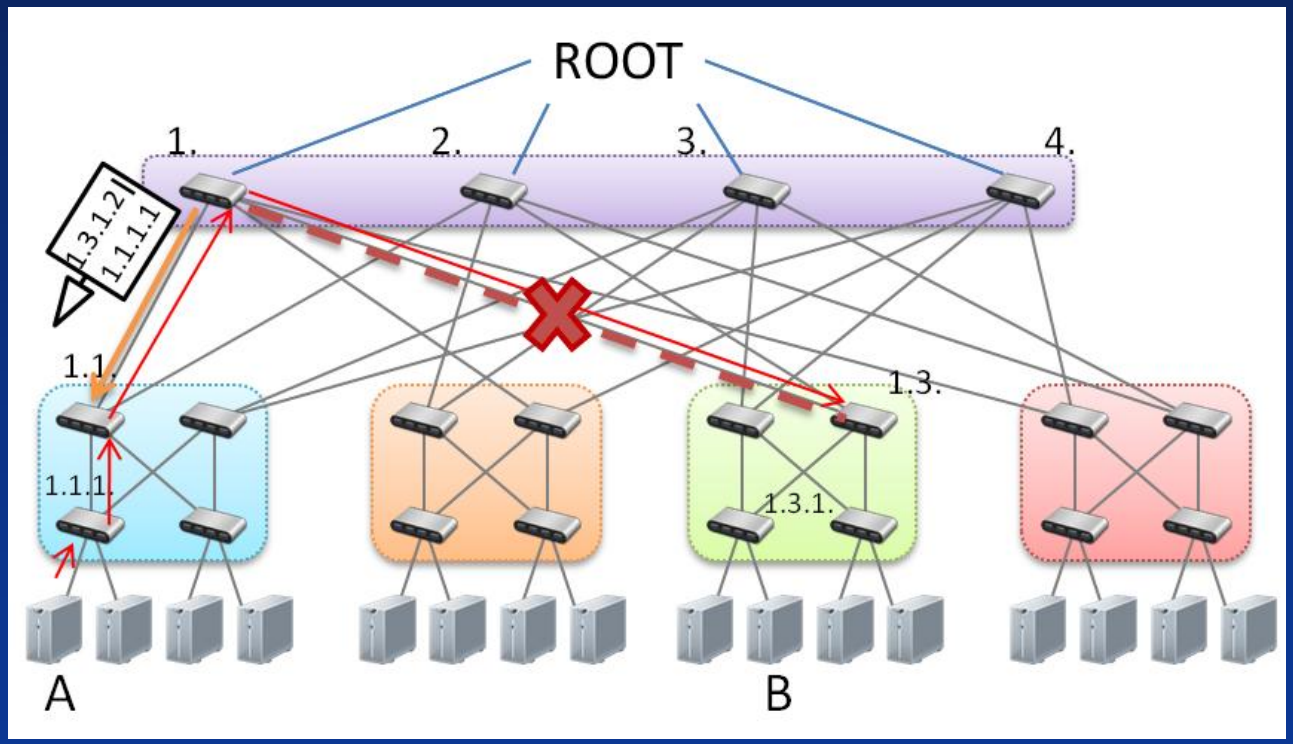
Protocol description

- *Tree-based path repair*
  - *Unicast*



Protocol description

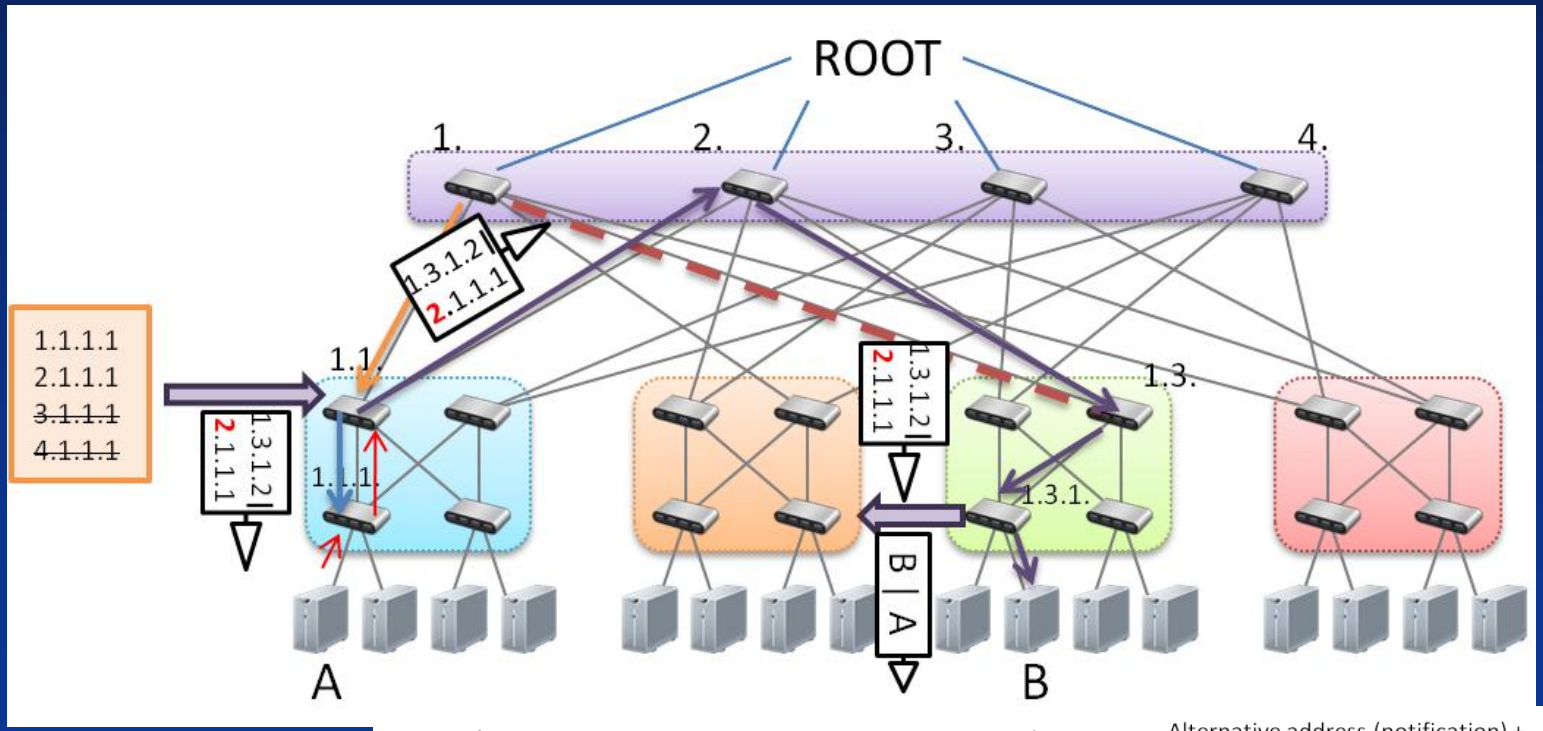
- *Tree-based path repair*
  - *Unicast*



  - *No possible duplicates, so **next common root** switch is chosen→bidirectional communication*

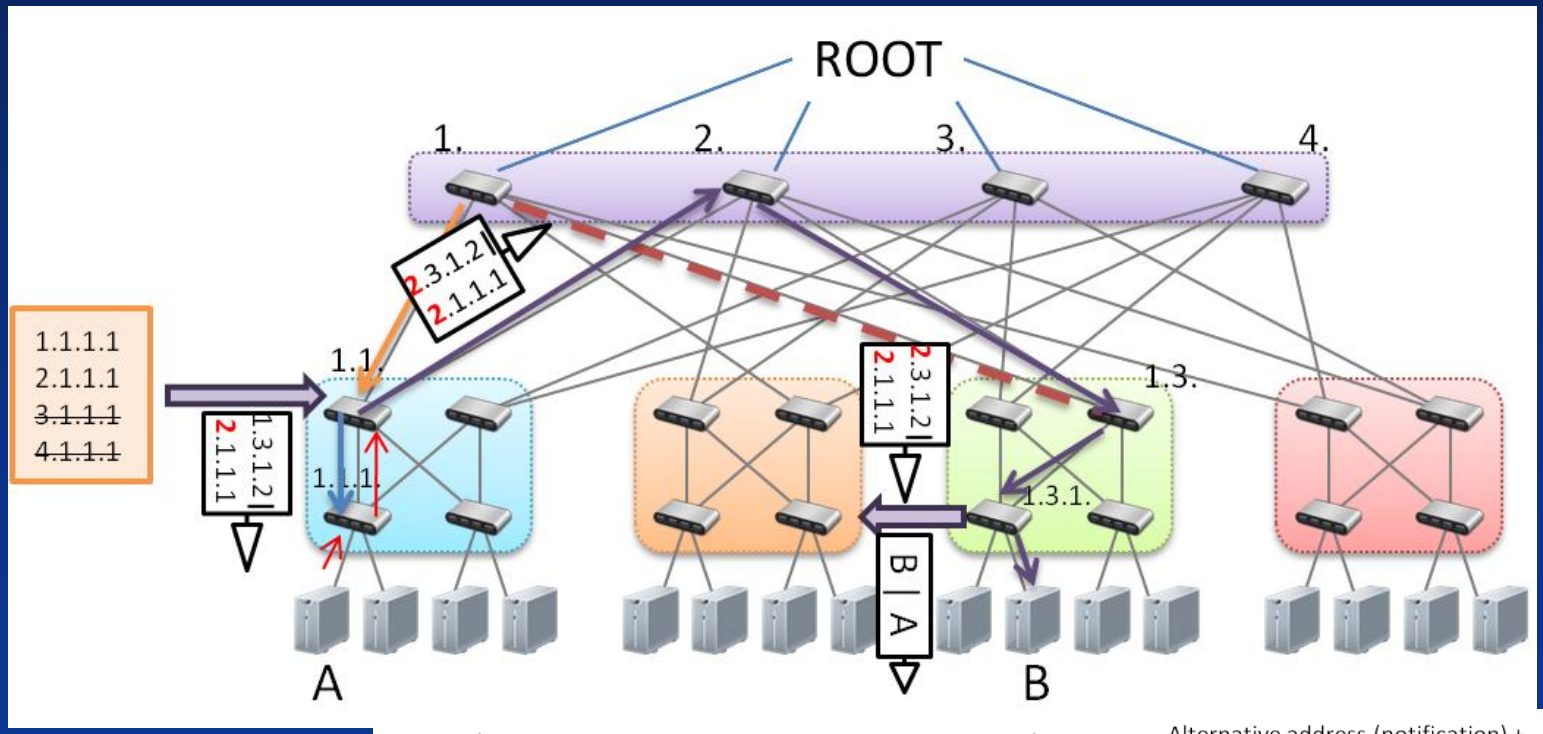
Protocol description

- *Tree-based path repair*
  - *Unicast → Frame + Destination notification + Source notification*



Protocol description

- *Tree-based path repair*
  - *Unicast → Frame + Source notification*



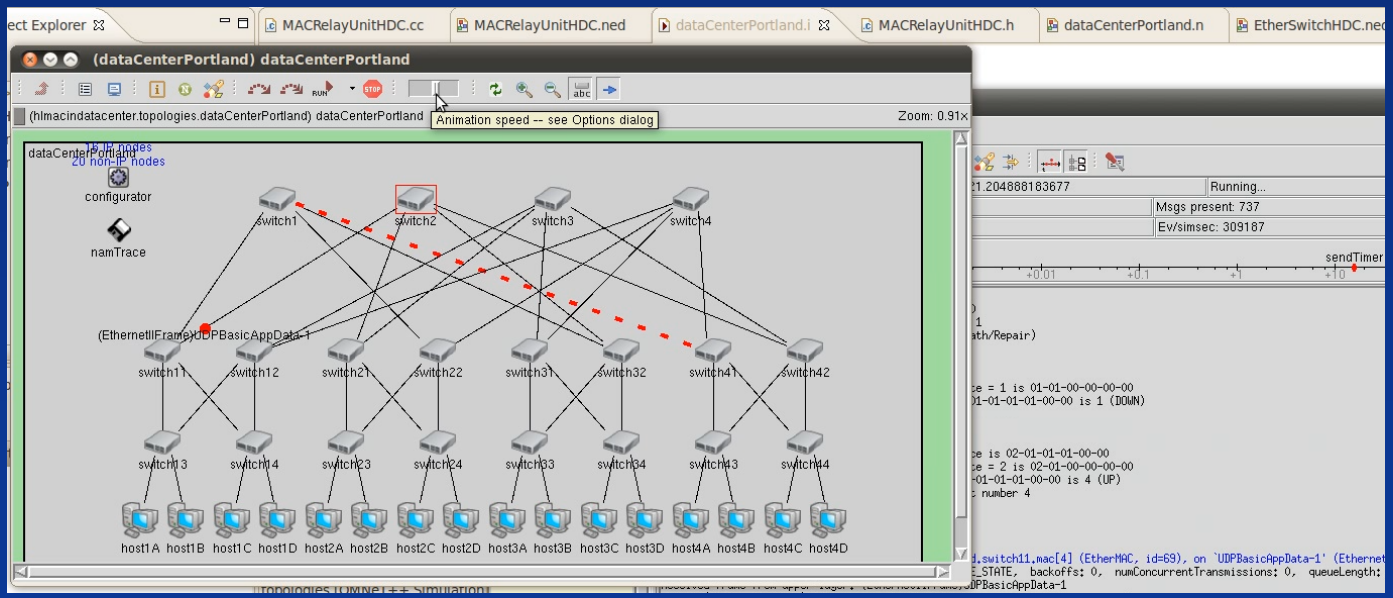
Evaluation

- *Simulation of Torii-HLMAC*
  - ***OMNeT++** (v4.1) ➔ Torii switch*

    *➔C++ implementation over **MACRelayUnit** (**inet** framework)*

    [Extended STP BPDU given as a parameter]
  - *PortLand topology + UDP traffic exchange*

    ➔Proven forwarding & path repair (different levels of link failure)



Ethernet Technology Summit

Evaluation

- *Use of Virtual Machines at hosts*
  - *Data center topologies: physical hosts usually composed by a number of **virtual machines** (VMs) installed*
  - *Torii only uses the **first 4 bytes** of HLMAC adddresses*

    *→So the **last 2 bytes** could be use to distinguish among those VMs (65535 active VMs), by being assigned in the reception order of their ARP messages.*
- *HLMAC Address Assignment Alternatives*
  - *In general, the Torii-HLMAC proposal takes 1 byte of the 6 of the HLMAC per hierarchical level, and 2 bytes for the VMs*

    *→Nevertheless, **fewer bits** could be assigned for this and could be used for some aditional functions (i.e. repair), without changing the protocol.*

Evaluation

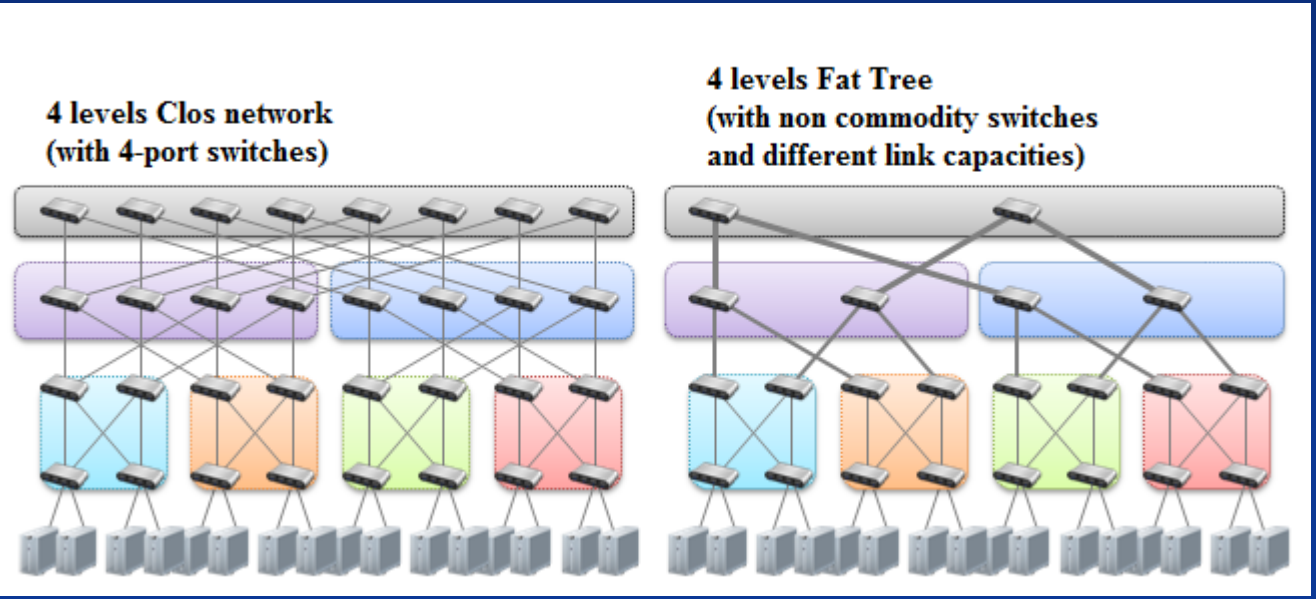
- *Inter-L2 Mobility*
  - *Gratuitous ARP propagates the new HLMAC information*
- *Generalization to any data center topology*
  - *We have just shown our proposel over the **PortLand topology**, what about different topologies?*
  - *The generalized PortLand topology will also work for Torii-HLMAC: << k-port switches can support 100 percent* throughput among $k^3/4$ *servers using* $k^2/4$ *switching elements and* the topology should be organized into *k pods, each* connecting $k^2/4$ *end hosts* >>

    *→ Torii-HLMAC could be used with **k up to 16**, more than enough.*

$k^2/4 < 2^6 \rightarrow k^2 < 64*4 = 256 \rightarrow$ **k < 16**

Evaluation

- *Generalization to any data center topology*
  - *While keeping the **pods**, any topology would work.*
  - The use of different topologies will depend on the most desirable feature:
    - less cost using cheap off-the-self components (Clos Network)
    - or less wiring complexity (Fat Tree).



Ethernet Technology Summit

Conclusions

- *Torii-HLMAC is a distributed, fault-tolerant, zero configuration fat tree data center architecture*
- *Forwarding needs **no tables***
  - *The only tables needed are the translations from MAC to HLMAC (and viceversa) of active hosts at the edge switches (table size <= active hosts)*
- ***On the fly** path repair*
- *No network manager*
- *No control messages*
- ***Load balancing** initially based on a hash function*
- *Hosts not affected (no need of any software or change)*
- *Independent of IP*

Conclusions

- ***Specific wiring*** *to be done at the construction of the topology*
- *Broadcast flooding is not avoided*
  - *ARP proxy could be used*
- *Multicast should be improved*
  - *So that not all the switches are broadcasted*

Ethernet Technology Summit

Conclusions

- ***Fat trees*** *are more convenient than Clos networks for Torii-HLMAC → simpler wiring*

- ***Deeper analysis*** *needed:*
  - *Comparison with other architectures*
  - *Setup time (Extended RSTP)*
  - *Broadcast reduction (proxys, host registration at directory, e.g. SEATTLE)*
  - *Multicast optimization (IGMP snooping, others)*
  - *Multiple path repair performance*

Ethernet Technology Summit

Torii-HLMAC

Thank you for your attention!

Any questions?

Elisa Rojas
elisa.rojas@uah.es
University of Alcala
(Spain)