GARR

The Italian Academic & Research Network

www.garr.it

Energy efficent MST in OpenFlow networks

European Workshop on Software Defined Network

L. Prete, F. Farina, M. Campanella, A. Biancini

Darmstadt, 26th Oct 2012

Consortium GARR

OUTLINE

- Motivation: OF, LearningSwitch module and loops
- The proposed solution
- Implementation and the virtual test-bed
- Measurements and validation
- Future works and conclusions

Fact 1: OpenFlow and modules

- OpenFlow is growing protocol with great potentials
  - Powerful but still too complex
  - Users experience difficulties in the use of modules

- The routing module is difficult and tricky
  - Powerful…
  - …but it requires in-depth knowledge of topology, devices, cluster, …

- The learning Switch is easier than Routing module. Useful to:
  - Experiment the power of the protocol (rules, actions, …)
  - Build and integrate new modules

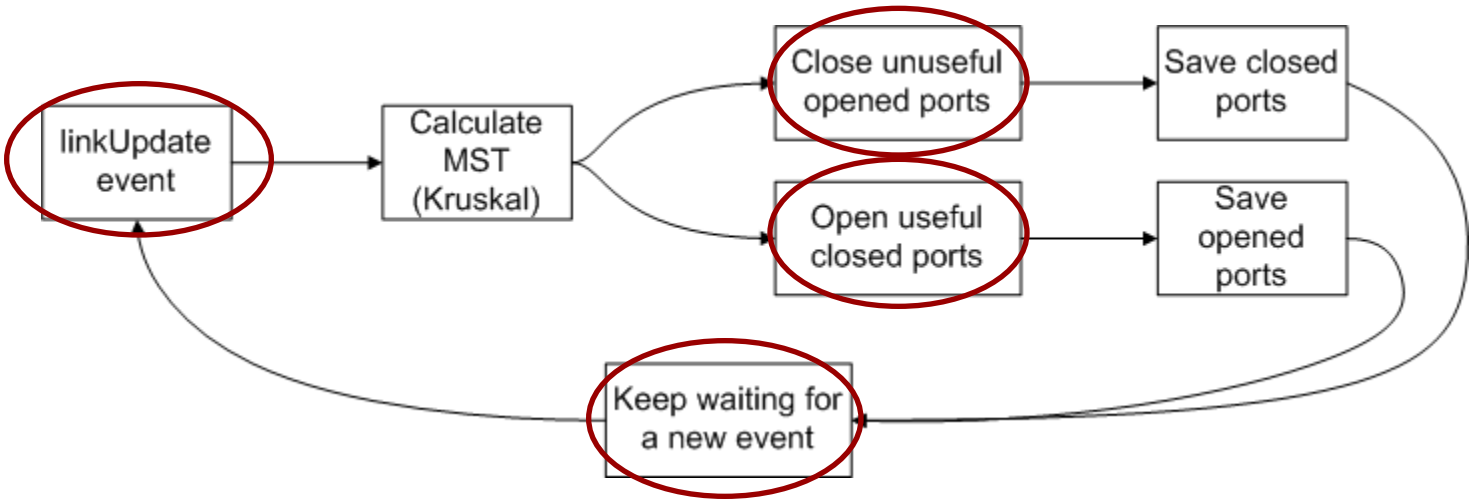
Fact 2:
OpenFlow and loop free technologies

- Fail-over mechanisms require topologies with physical loops
- Without loop-free technologies broadcast storms
  - ARP requests, any broadcast or multicast based protocol, …
- “Common” L2 switches use the Spanning Tree Protocol
  - Software OF switches should export the STP_bit, but most of them DON’T!
- Problem:
  - Lack of a loop free technology using OF learning switch module

From a need to the idea

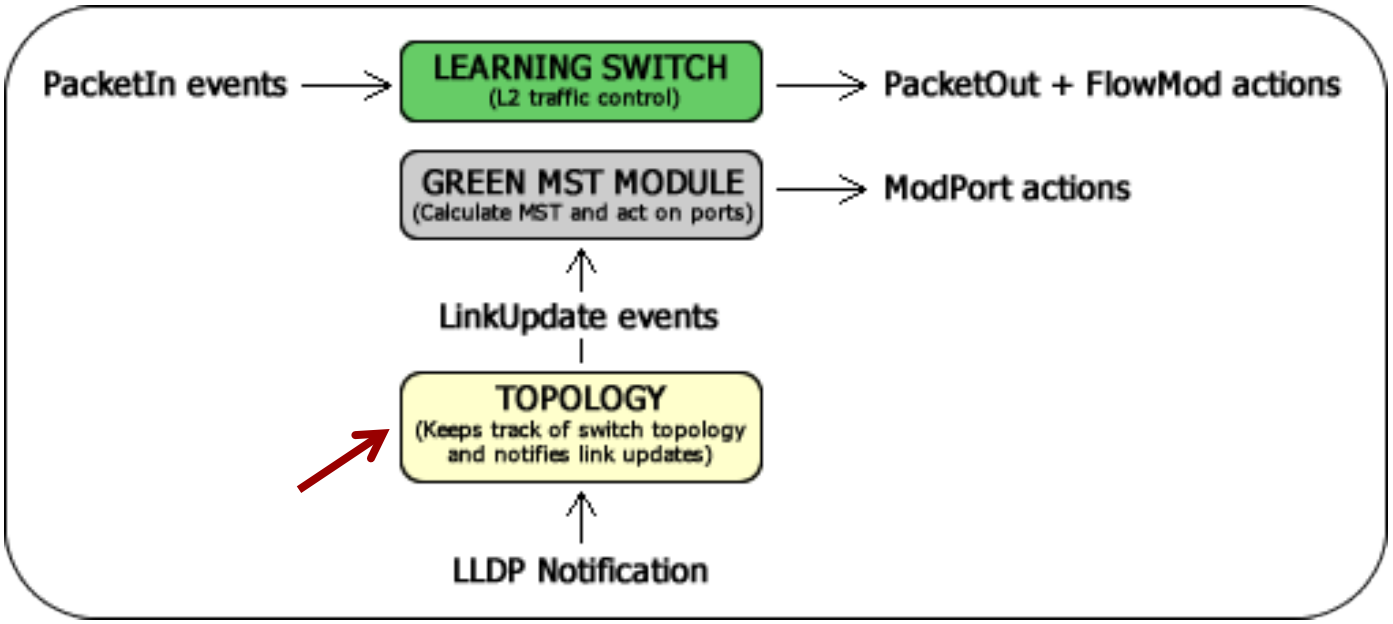
- Need
  - LearningSwitch + looped topologies + non-STP switches
  - Simple Plug&Play solution for beginners
- Idea
  - Build the Minimum Spanning Tree of the network
  - Turn off ports outside MST
- Goal
  - Prevent broadcast storm
  - Offer failover mechanisms
  - Save energy

GreenMST: how it works



- Link between switches changes: linkUpdate event
- After each topology change recalculate Minimum Spanning Tree
  - MST is computed using the Kruskal algorithm
- Ports status is modified
  - Open ports not included in the MST are closed
  - Closed ports included in the MST are opened

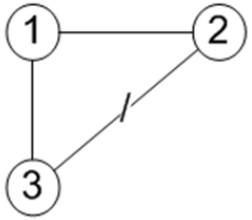
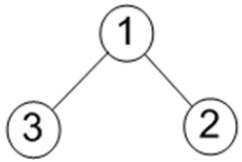
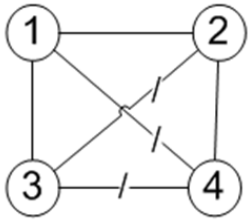
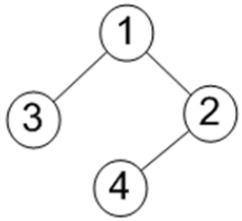
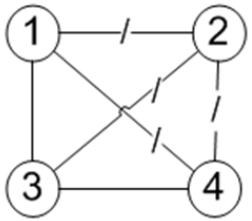
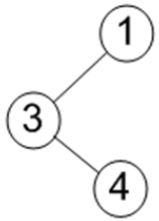
GreenMST: the proposed solution



Modular approach

- Topology keeps the switch map updated and it sends updates to the Green MST module when a link changes
- Green MST computes the new MST using current topology information and it sends ModPort actions to the switches
- LearningSwitch module continues to work normally, sending packets only through the opened ports

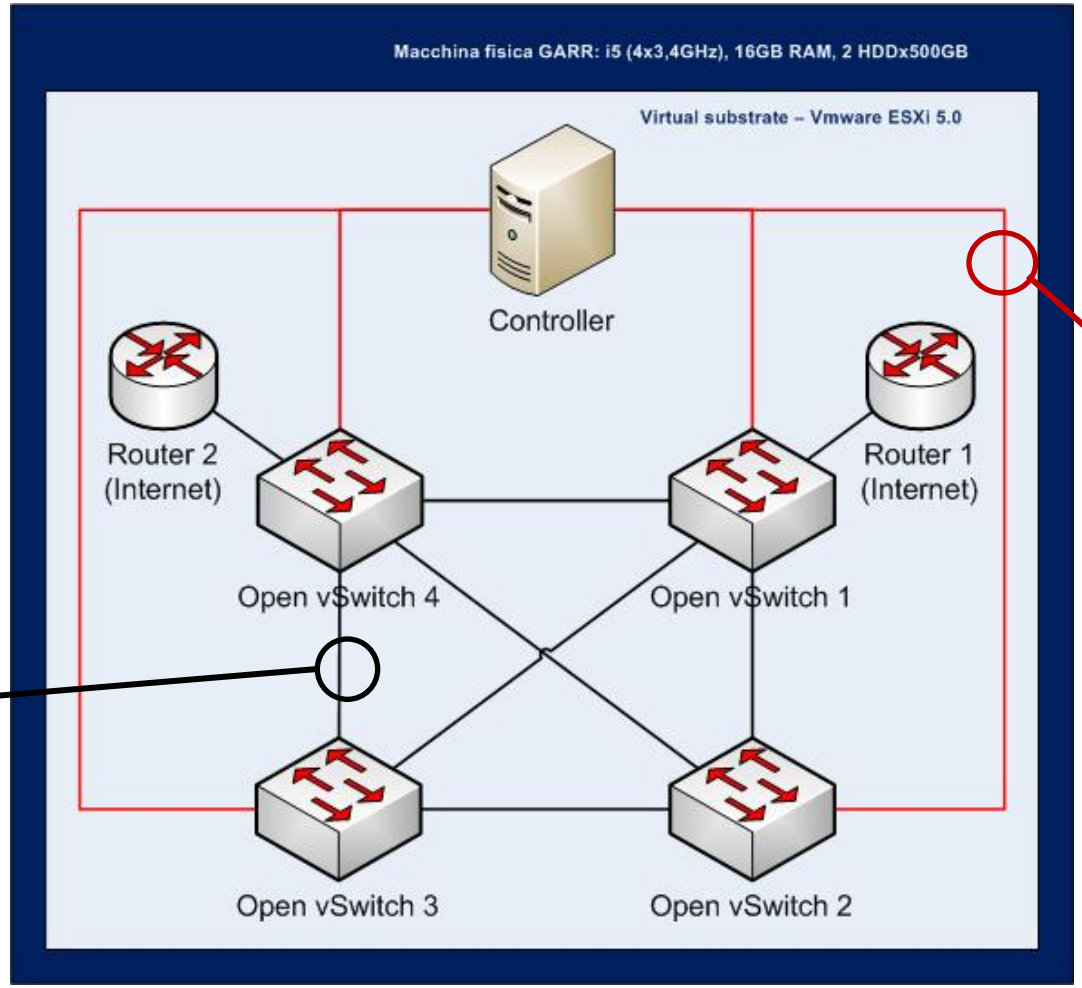
MST computation and Change Set

	Topology	MST	Change Set
t_0			Add: Ø Remove: Ø
t_1			Add: 2—4 Remove: 1—4 3—4
t_2			Add: 3—4 Remove: 1—2 2—4

Saved 3-2 closed

- Ports status saved in a data structure
  - Minimization of ModPort commands to the switches

Implementation: the virtual testbed



- **Goal:** experiment OpenFlow features and potentialities

Implementation: Software used

OpenvSwitch

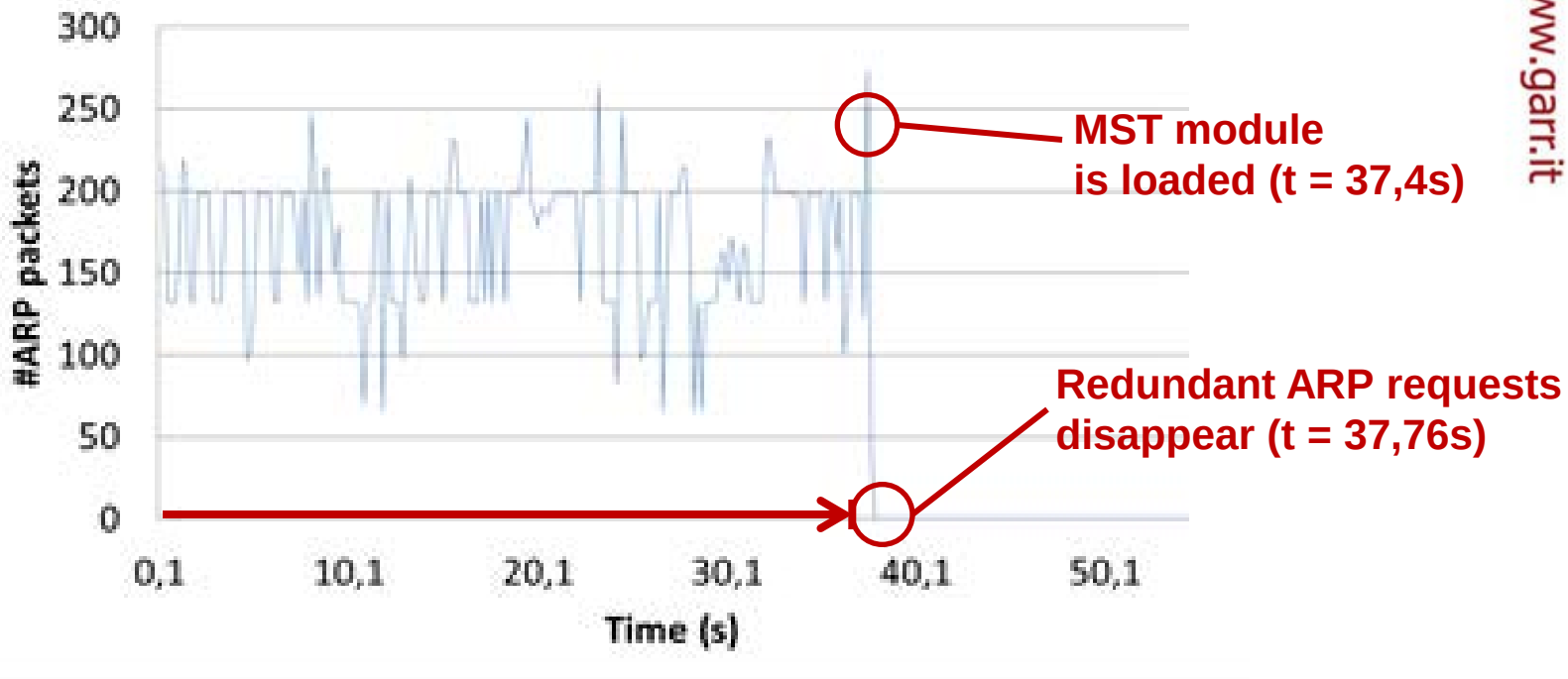
- Enterprise functionalities
- Used by different hypervisors, es. Xen

+

Beacon controller

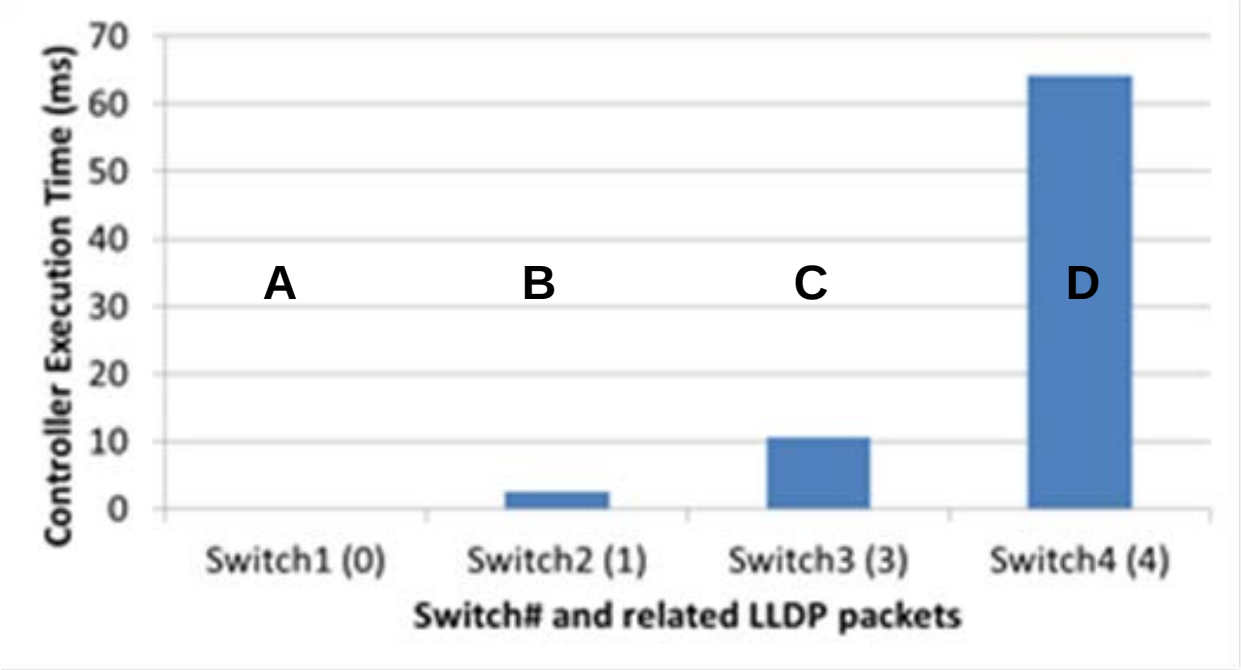
- Java
  - Multi-platform
  - Thread-oriented
- Used in large testbed (>150 switch)
- Modular

Experiment 1 – ARP request



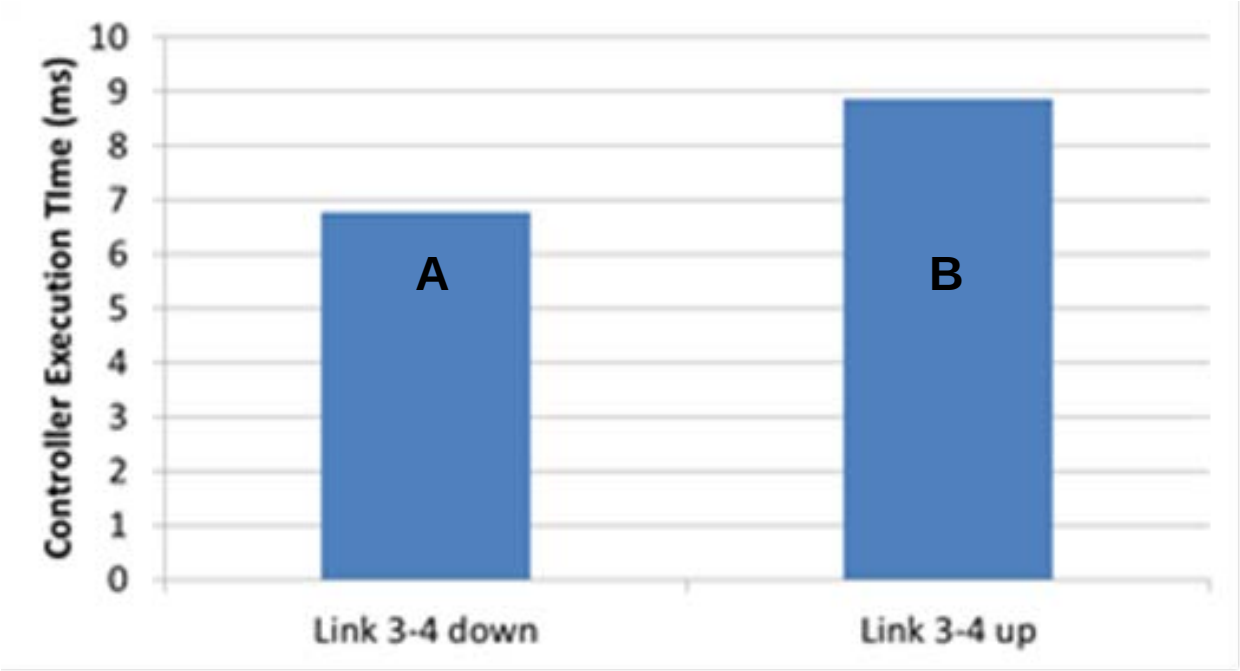
- From the point of view of OVS01 (one of the switches) after a PING
  - $[t_0 - t_{(37,3)}]$ => observation window without MST module
  - $t_{(37,4)}$ => MST module is loaded
  - $t_{(37,76)}$ => redundant ARP requests disappear
- Final MST + ModPort actions end after 0,36 sec

Experiment 2 – switch activation



- From the point of view of the controller
  - t_0 => GreenMST module is loaded and MST is calculated
    => No switches are active
  - $t_{A,B,C,D}$ => Switches are activated
    => MST is re-calculated and ports status is modified (except for OVS01)

Experiment 3 – Link status change



- From the point of view of the controller
  - t_0 => Network is stable and MST is already calculated
  - t_A => Link between OVS03 and OVS04 is deactivated
    - New MST calculated in **7 ms**
  - t_B => Link between OVS03 and OVS04 is re-activated
    - New MST calculated in **9 ms**

Conclusions and future works

- Now
  - Enable loop free technologies and failover mechanisms for Learning Switch OF modules
    - As well if the devices do not support the STP technology!
  - Save energy turning dynamically off the unused ports of the switches

- In the future
  - Assign dynamically a cost to the links between the switches using link parameters (bandwidth, delay, jitter)
    - ...but also SNMP or passive flow monitoring protocols (Netflow, sFlow, ...)
  - Memorize the alternative paths between two nodes before a fail
    - After a fail the alternative path is chosen
  - Integration with the Routing module

THANK YOU

Question

Get the code
https://github.com/LucaPrete/GreenMST

Reference

- Project page: https://github.com/LucaPrete/GreenMST
- OpenFlow official site: http://www.openflow.org
- Open vSwitch: http://www.openvswitch.org
- Controller
  - Beacon: https://openflow.stanford.edu/display/Beacon/Home