Multi-homed network : Changelog

Group 3 :
Romain Dizier, Quentin Gusbin, Thibault Jacques,
Léonard Julémont, Nicolas Vrielynck

May 1, 2017

1 Introduction

This document presents the various elements that have been modified or added since the first submitted version of the project.

2 Topology

1. PYTH-CARN link moved between PYTH and HALL. It allows source routing of the traffic between PYTH and HALL to work even the primary link fails. It also avoids the traffic between the two data centers to go around the entire network in case of the failure of the primary link.

2. Modification of addresses between routers. The link number now starts at 1 instead of 0 to allow the future use of number 0 for loopback addresses.

3 Routing

1. Only accept default route over BGP from ISPs on HALL and PYTH. It makes it possible to avoid filling the BGP table with unnecessary routes coming from the providers.

2. Add scripts on HALL and PYTH to check the status of the primary link. If it fails each the script on each router modify the route used by the source routing to use the backup link. The script switches back to the primary link when it becomes available.

3. Add new test which checks that the network still behave correctly when the primary link between HALL and PYTH is down.

4 End user management

1. Add the RDNSS extension to cover more users.

2. Some modification about the directories to find the different files

    - Use the standard name of the directory in the simulated nodes.

- Put every directory about the end user management in one directory at the root of the project

3. Modification of the values of the SOA resource record and explanations about it.
4. New tests added for the DNS

5 Security

1. All the scripts for the different firewalls and the script to start these firewalls are now in the single directory "ip6tables".
2. Added variables for often used values at the top of all the firewall scripts.
3. Added several rules for the OSPF protocol.
4. Modified the ordering of ip6tables rules : instead of mixing "Append" (-A) with "Insert" (-I), only "-A" is used for readability.
5. Added several loops in the firewall scripts to avoid redundancy with rules where only one parameter vary (mainly for the allowed tcp and udp traffic).
6. Modified policies for traffic related to HTTP (port 80), HTTPS (port 443), SSH (port 22) and SMTP (port 25).
7. Merged INPUT rules for echo request (icmpv6 type 128/0) to avoid redundancy over limit rate.
8. Removed incorrect set of rules for traffic coming from "known and trusted sources"
9. Merged ICMPv6 rules together to avoid duplication and allow error or info messages to be sent back if needed.
10. Modified and moved logging policies to avoid recording packets that are neither dropped nor needed.
11. Added policy for DHCPv6 traffic (due to the servers used in parallel of SLAAC).
12. Added policy for iperf3 packets (used in the "Quality of Service" part of the project).
13. Added policy for BGP packets at the two border routers.
14. Added rate limitation for Neighbor Solicitations packets (icmpv6 type 135/0) to avoid DoS attacks on neighbor cache.
15. Modified policy for unsolicited Router Acknowledgement packets coming from students.
16. Modified the script for loading and setting the firewalls when the network starts ("startFw.sh" in the ip6tables directory).

    Possible improvements : implementation of RA Guard and SEND against spoofing attacks.

6 Quality of service

1. Added an ssh class so that we can still connect to our servers/ routers in case of congestion. Before this change ssh traffic was classify as *other traffic* and had a really low priority.

2. Explained more precisely in the report why we are using pfifo and not sfq for the three first classes. As those classes are never congested a sfq is useless and takes much more time to process packets.

3. Changed wrong classid in the tree in the report for *other traffic* class.

4. Used version 3.1 instead of 3.0 of iperf3. This brings the new argument –one-off that stops the iperf server after one test. This way the server doesn't need an iperf server open continuously and can simply start one before the test and shut it down afterwards.