Security Labs in OPNET
IT Guru

Enginyeria i Arquitectura La Salle
Universitat Ramon Llull
Barcelona 2004
Security Labs in OPNET IT Guru

Authors:

Cesc Canet

Juan Agustín Zaballos

Translation from Catalan:

Cesc Canet
Overview

This project consists in practical networking scenarios to be done with OPNET IT Guru Academic Edition, with a particular interest in security issues.

The first two parts are a short installation manual and an introduction to OPNET. After that there are 10 Labs that bring into practice different networking technologies. Every Lab consists in a theoretical introduction, a step-by-step construction of the scenario and finally Q&A referring to the issues exposed.

**Lab 1: ICMP Ping**, we study Ping traces and link failures.

**Lab 2: Subnetting and OSI Model**, we study tiers 1, 2 and 3 of the OSI model, and the Packet Analyzer tool to observe TCP connections.

**Lab 3: Firewalls**, we begin with proxies and firewalls. We will deny multimedia traffic with a proxy, and study the link usage performance.

**Lab 4: RIP** explains the RIP routing protocol, and how to create timed link failures and recoveries.

**Lab 5: OSPF** compares RIP. We study areas and Load Balancing.

**Lab 6: VPN** studies secure non-local connections. A Hacker will try to access into a server that we will try to protect using virtual private networks.

**Lab 7: VLAN** creates user logical groups with Virtual LANs. Studies One-Armed-Router interconnections.

**Lab 8: Dual Homed Router/Host**, **Lab 9: Screened Host/Subnet. DMZ** and **Lab 10: Collapsed DMZ** explains the static routing tables, ACLs, proxies and internal vs. perimetric security. Lab 10 is 100% practical, we want you to create it on your own, a piece of cake if you did the other Labs!
Lab 5: OSPF

OSPF (Open Shortest Path First) is yet another routing protocol. In a nutshell:

- It is a Link-State protocol instead of a Vector-Distance as RIP. Link-State protocols converge faster than Vector-Distance.
- It is an Interior Gateway Protocol (IGP), used for Internet routing.
- Based on Dijkstra algorithm.
- Based on IP, it does not use any upper layer service (RIP uses TCP/UDP).
- Can calculate many routes for each IP service.
- Can use Load-Balancing (balanced distribution of traffic on different routers) for same-cost routes. If there are many same-cost routes, OSPF will write all them up in the routing tables.
- The effective bandwidth is reduced by the protocol because of its multicast IP messages usage, instead of broadcast. Does not overload those systems that do not use OSPF.
- Can use many metrics: outgoing frame length, throughput, delay, link liability, cost, etc. RIP can only use hop counting.
- Messages are sent only if there are changes on the topology.
- Needs a better router processing speed: routers have a link state table and a route state table as well.
- Can implement classless addresses (CIDR).
- Can use areas. The traffic generated by the OSPF protocol can be significantly decreased dividing the network in areas. Routers do not have to save any other data but the routing within their own area and a route to other areas. This reduces a lot the OSPF load on the network and the global results can be better than with a more accurate route. Areas are identified by an Area ID. The Area ID is a 32-bit number using A.B.C.D notation (similar to IP addresses). The backbone is usually identified by 0.0.0.0. The other areas use 0.0.0.1 and so on.
- OSPF is based in open standards and public domain specification. The current implementation is based on RFC 2328.
Lab Description

This Lab is divided in three scenarios: **NoAreas**, where no area is defined. Its proves how OSPF chooses the cheapest area when two possible paths exist; **WithAreas** explains areas, and **LoadBalancing** shows how two equal-cost routes are used equally to divide the traffic across the route, and thus reducing the total load on each link.

Scenario Creation

1. Open a new OPNET IT Guru Academic Edition Project (**File** → **New Project**) using the following parameters (use default values for the remainder):

   - **Project Name**: `<your_name>_OSPF`
   - **Scenario Name**: **NoAreas**
   - **Network Scale**: **Campus**
   - **1000x1000 Meters**

   Press **Next** several times until the Startup Wizard finishes. A new blank grid is now created in the Project Editor.

2. Deploying devices on the grid:
   In the Project Editor, **Topology** → **Rapid Configuration**, choose **Configuration**: **Mesh, Full** and click **OK**. Set the following parameters in the dialog:

   ![L5.1 Rapid Configuration](image)

   This creates 5 ethernet_slip_8_gtwy nodes in a full-meshed, star-topology.
Deploy the remaining nodes as in picture L5.2. Zoom + before to work more comfortably. All nodes are type ethernet_slip8_gtwy and are connected using PPP_DS3 wiring.

Use the same node names: we will refer to them using these names hereinafter. We used OPNET default names (including autonumbering). Use the same order of creation to obtain the same numbers.

The picture shows the links cost as well. We will explain how to set the costs later, but it is not necessary to use the labels on the links (Topology→Open Annotation Palette). The colored arrows have a special meaning, as we will see.

![L5.2 The completed scenario](image)

3. Configuring the link cost:

The link cost formula is given by: \( \text{Cost} = \frac{\text{Reference Bandwidth}}{\text{Link Bandwidth}} \). The default value is Reference Bandwidth = 1,000,000 Kbps.

To assign a link cost (for instance, to assign 10 units to node 8 – node 9):

\[
10 = \frac{1,000,000}{\text{Link Bandwidth}} \Rightarrow \text{Link Bandwidth} = 100,000
\]

The Link Bandwidth value can be set to a specific link by selecting all the same-cost links (hold Shift key + Left button for multiple selections), and then

Protocols→IP→Routing→Configure Interface Metric Information.
For the example, we input **Bandwidth (Kbps): 100000**, checked **Interfaces across selected links** and pressed **OK**. The remaining parameters use default values.

**5.3 Setting the link metrics**

The remaining links have Cost=1 $\to$ Bandwidth = 1000000 Kbps.

4. Configuring traffic demands:
   
   We create two traffic demands
   
   - *node_7* to *node_9*
   - *node_8* to *node_9*

   To create the *node_7* to *node_9* demand (the other one is similar)
   
   - Select *node_7* and *node_9*.
   - In the Project Editor, **Protocols**$\to$**IP**$\to$**Demands**$\to$**Create Traffic Demands**... Select **From**: *node_7*, and pick up a color for this traffic demand (this color will be used in the Project Editor map). Choose **White** and click on **Create**.

**L5.4 Creating traffic demands**

We create another Traffic Demand from *node_8* to *node_9*, in yellow.
Traffic demands can be hidden/shown at any moment with View→Demand Objects→Hide All/Show All.

5. Setting addressing and routing protocols:

- Go to Protocols→IP→Routing→Configure Routing Protocols... without having any node selected. A new dialog will pop up as seen in picture L5.5. At the routing protocols checkbox, we mark only OSPF, and select as well Visualize Routing Domains and All Interfaces (including loopback). The remaining options will be left with default values. Click OK.

![Routing Protocol Configuration dialog]

**L5.5 Setting the protocol at the routers**

- Select the Traffic demand origin nodes (node_7 and node_8), and at the Project Editor go to Protocols→IP→Routing→Export Routing Table for Selected Routers and press OK at the confirmation dialog. This way we can study the routing of the packets at the end of the simulation.

![Routing Table Export dialog]

**L5.6 Confirming the Routing Table Export**

- Now we can assign IP addresses to all routers at the Project Editor, Protocols→IP→Addressing→Auto-Assign IP Addresses.

6. Setting up the simulation:

At the Project Editor click at configure/run simulation, and write Duration: 10 minute(s), and press OK (don’t start the simulation yet).
Creating the second and third scenarios

At the scenario we have just created, all the routers belong to the same hierarchical level (area). We have not enforced Load Balancing on any route. We can now create the second and third scenarios: the second will be WithAreas and it will define three areas and a central backbone. The third scenario allows Load Balancing within nodes 7 and 9.

1. Creating the scenario WithAreas

- At the Project Editor, select Scenarios→Duplicate Scenario..., and call it WithAreas. Press OK.
- The first area is 0.0.0.1. In order to configure it, select it with the Shift key the links connecting nodes 4, 5, 6 and 7, and then go to Protocols→OSPF→Configure Areas and assign the value 0.0.0.1 to the Area Identifier, and press OK.
- Do the same for the second area (0.0.0.2), with the links connecting nodes 0, 8 and 9.
- The third area is 0.0.0.3 and is composed by the links that connect nodes 1, 10 and 11.
- The central backbone is not configured because this is done by default. It’s Area ID is 0.0.0.0.
- By doing Protocols→OSPF→Visualize Areas... we can show the links of the different areas expressed in different colors.

L5.7 Dividing areas in different colors
L5.8 The scenario whit areas in colors

2. Creating the scenario LoadBalancing

- At the Project Editor, Scenarios → Switch to Scenario → SenseArees. We use this scenario so that now we can make changes on the duplicate scenario, without having any areas.
- Select Scenarios → Duplicate Scenario... and call it LoadBalancing. Click OK afterwards.
- Select nodes 7 and 9 holding the Shift key.
- Protocols → IP → Routing → Configure Load Balancing Options. We must ensure Packet-Based is selected and the radio button Selected Routers is checked. Click OK.

L5.9 Configuring Load Balancing

Edit the Attributes of node_9, and set the OSPF value: Parameters → Loopback Interfaces → row 0 → Area ID: 0.0.0.2.
L5.10 Configuring the Area ID of the loopback

3. Running the simulation:
   From the Project Editor, **Scenarios**→**Manage Scenarios**...
   Change the values on Results row to **<collect>** or **<recollect>** in all scenarios.
   When we now press **OK** the 3 simulations will run, one after another.
   When the simulations are finished, click on **Close**. At the top left part of the dialog we can see a field called **Simulation runs to go**, that needs to be zero when the simulations are finished before we can click **Close**.

L5.11 Details of the dialog

Results analysis

1. **WithoutAreas**
   - At the Project Editor, **Scenarios**→**Switch to scenario**→**WithoutAreas**
   - To see the traffic demand between 8 and 9, from the Project Editor go to **Protocols**→**IP**→**Demands**→**Display Routes for Configured Demands**.
     Drop down the branches completely and set **Display**: **Yes**, and click on **Close**. Now we can see the traffic demand.
L5.12 Configuring the demand visualization

- Repeat the last step so we can see the traffic between 7 and 9. Depending on the order the links were created, it will pass through router 6 instead of the router 5. Both two paths have the same cost. OPNET will choose one depending on the order. (Q1)

WithAreas

- From the Project Editor, Scenario→Switch to Scenario→Areas
- See the route between 8 and 9. (Q2)

Load_Balancing

- At the Project Editor, Scenario→Switch to Scenario→LoadBalancing
- See the routing between nodes 7 and 9. We did load balancing, so the both minimum cost paths are highlighted. They both cost the same.

Questions

Q1 Which routes are traffic demands node_8→node_9 and node_7→node_9 following at WithoutAreas? Why?

Q2 Which route is taking the traffic demand node_8→node_9 at WithAreas and why?

Q3 Which route is taking the traffic demand node_7→node_9 at LoadBalancing and why?
Answers

Q1 The result can be seen at picture L5.13

It will take the shortest path, but not the shortest. Regarding to the node_8 → node_9 demand, RIP would have used the link node_8 → node_9, but OSPF is using the links 8-0-1-9. There are two possible paths for the other demand, the one that goes through node_5 and the one that goes through node_6. OSPF will take one of them for the load balancing. In this case, it takes node_5.

Q2 We can see now it is not taking the cheapest path, but the cheapest in its own area. When we do areas, each router has a route to all possible destinations in its own area, and a gateway to the remaining stations.
Q3 As we have said at Q1, there are two possible paths to communicate each other node_7 and node_9. LoadBalancing will take both of them, dividing the load equally for the two routes.

L5.15 Route with load balancing