# Building the Network Management topology

## Contents



We're going to build a topology we use in the Network Management class (and some others). This will require to have some routers on the backbone, and behind each router, 4 PCs will be started.

The topology will look as follows:

[pic]

Let's get started! You will be working as a group - this means you will cooperate on this task.

# 1 Building the router class

## 1.1 First, do a quick check that the VMs are not running, or other routers.

- use `virsh list` to identify the machines you will need to shutdown
- if you are running screen, reconnect to that, and stop dynamips and dynagen

Check using `ps ax` that no dynamips or dynagen processes are left running.

### 1.2 Now, let's make sure the workshop kit repository is up to date:

```
cd /home/nsrc/workshop-kit
git pull
```

### 1.3 Now, run the `vm_servers.yml` playbook again, but we're only going to

modify the DHCP server

```
cd ansible
sudo ansible-playbook -t dhcp vm_servers.yml
```

If everything went well, a new file should have been installed, `/etc/dhcp/dhcpd.nmm.conf`. Take a look at this file. You will notice there are 9 groups of 4 PCs. That's a total of 36 machines we will be starting!

To be on the safe side, we will run:

```
sudo ansible-playbook -t interfaces networking.yml
```

And to be certain all the interfaces have started, run the following commands:

```
sudo ifup br1 br2 br3 br4 br5 br6 br7 br8 br9 tap1 tap2 tap3 tap4 tap5 tap6 tap7 tap8 tap9
```

Ignore the complaints if it says the interfaces are already up.

### 1.4 Let's grab the `class.net` file for the Network Management topology.

In the future, we'll probably use ansible, but in the meantime, we've got a standard version of the topology ready for you to download!

You can copy it from the workshop-kit repository (under workshop-kit/conf/dynagen/nmm/class.net).

Create a working directory for the nmm class:

```
cd /home/nsrc
```

You should already have a work directory for dynamips, but just in case:

```
mkdir -p dynamips/nmm
cd dynamips/nmm
```

Copy the `class.net` file here:

```
cp /home/nsrc/workshop-kit/conf/dynagen/nmm/class.net .
```

If the file cannot be found, then you need to run `git pull` to update the workshop-kit repository!

Alternatively, copy-paste the `class.net` file from

```
http://wsnoc.nsrc.org:8000/nsrc/workshop-kit/raw/master/conf/dynagen/nmm/class.net
```

... into the `class.net` file.

Note: You MUST clean the contents of the cache directory, as was explained in class. If in doubt:

```
rm -f /home/nsrc/dynamips/cache/*
```

Note: it's important to do the cleanup!

## 1.5 Copy the base router configuration files

We still need to load some basic configuration into the routers, otherwise they will start with no IP address configured!

We'll copy the `lab-base` configuration from the workshop-kit:

```
cd /home/nsrc/dynamips/nmm
mkdir work
mkdir configs
cp /home/nsrc/workshop-kit/conf/dynagen/nmm/configs/lab-base/* configs/
```

## 1.6 Now, we're ready to start the routers.

```
cd /home/nsrc/dynamips/nmm
```

We'll use `screen` so we can have `dynamips` running in one screen, and `dynagen` in the other. You should know how to do this!

Once `screen` is started, start `dynamips` by running the following command:

```
/home/nsrc/workshop-kit/conf/dynagen/run-dynamips
```

You should see the following output:

```
Cisco Router Simulation Platform (version 0.2.10-amd64/Linux stable)
Copyright (c) 2005-2011 Christophe Fillot.
Build date: Nov  2 2013 19:03:02

Local UUID: 4b9d715c-b9da-4939-ac35-d7821b5b4be2

Hypervisor TCP control server started (port 7200).
```

Make sure the process is running:

```
ps ax | grep dynamips
```

Switch to/create another `screen` windows and we are now ready to start the `class.net` using `dynagen`:

```
cd /home/nsrc/dynamips/nmm/
```

Launch `dynagen` on the `class.net` file:

```
dynagen class.net
```

All routers should start, and you should see:

```
Reading configuration file...

Network successfully loaded

Dynagen management console for Dynamips and Pemuwrapper 0.11.0
Copyright (c) 2005-2007 Greg Anuzelli, contributions Pavel Skovajsa

=>
```

Run the `list` command to see that all routers are up. You should see:

```
Name       Type       State      Server          Console
r1         7200       running    s1.ws.nsrc.org:7200 2101
r2         7200       running    s1.ws.nsrc.org:7200 2102
r3         7200       running    s1.ws.nsrc.org:7200 2103
r4         7200       running    s1.ws.nsrc.org:7200 2104
r5         7200       running    s1.ws.nsrc.org:7200 2105
r6         7200       running    s1.ws.nsrc.org:7200 2106
r7         7200       running    s1.ws.nsrc.org:7200 2107
r8         7200       running    s1.ws.nsrc.org:7200 2108
r9         7200       running    s1.ws.nsrc.org:7200 2109
```

## 1.7 Import the router configurations

Now, time to import the router configurations for r1 to r9.

Run this at the `dynagen` prompt:

```
=> import /all configs
```

You should see:

```
Importing r2 from r2.cfg
Importing r5 from r5.cfg
Importing r8 from r8.cfg
Importing r1 from r1.cfg
Importing r6 from r6.cfg
Importing r7 from r7.cfg
Importing r4 from r4.cfg
Importing r9 from r9.cfg
Importing r3 from r3.cfg
```

Reload all the routers so the new configs are loaded:

```
=> reload /all
```

Wait a few minutes - you should be able to ping the routers:

```
r1 -> 10.10.0.1
r2 -> 10.10.0.2
...
r9 -> 10.10.0.9
```

Also, you should be able to telnet to each router, using the following credentials:

```
user: cisco
password: cisco
```