IP Routing: dynamic routing

Part I: Dynamic routing using OSPF

In the previous set of labs, you manually configured static routes on each of your group's routers. We did this for two reasonss:

- to avoid the "single point of failure" when exchanging traffic between groups
- to make the traffic go faster (by avoiding the GW)

But as you saw, it was a manual process: it was time consuming, and it has to be repeated every time a new network (and router) is added or removed from your network. And these changes will need to be performed on ALL routers:

Routers on net (configs to change)	Total changes
2	2
3	4
4	6

(T = R * 2 - 2)

What if we could just tell the routers to talk to each other, and have them exchange information about the networks they know about ?

Well, this is what the Open Shortest Path First protocol (OSPF) is for.

We'll be enabling OSPF on each of the routers in the class.

1. Log in to your router

```
rtrX> enable
Password:
rtrX#
```

- Make sure your static routes are still in place for the other groups (this is what you have done in the previous lab)

```
rtrX# show ip route

S       10.10.2.0 [1/0] via 10.10.0.2
S       10.10.3.0 [1/0] via 10.10.0.3
...
S       10.10.9.0 [1/0] via 10.10.0.9
```

The 'S' flag is telling us that the routes are statically defined.

2. Let's enable OSPF on our routers!

- First, let's tell our router it needs to run OSPF:

```
Router1# conf terminal
```

- Now we're telling our router to create an OSPF process:

```
rtrX(config)# router ospf 100
```

- The next command is going to tell the router that by default we do NOT
  want to enable OSPF on all interfaces by default. We will enable this
  only for the interfaces we want (case by case)

```
rtrX(config-router)# passive-interface default
```

- Now we tell the OSPF router process that interface fastEthernet 0/0
  should be "speaking" OSPF with the other routers on the backbone.
  Why fastEthernet 0/0 ? Because it's the interface that connects your
  router to the backbone.

```
rtrX(config-router)# no passive-interface fastEthernet 0/0
```

- The next step is to tell the OSPF router process that the networks
  we are connected to (that is, your network - 10.10.X.0/24) should
  be announced to other routers speaking OSPF.

```
rtrX(config-router)# redistribute connected subnets
```

- The last step is to tell the OSPF process which OSPF area to join on
  the backbone interface fastEthernet 0/0 (we'll mention what an Area
  is, in class)

```
rtrX(config-router)# exit
rtrX(config)# interface fastEthernet 0/0
rtrX(config-if)# ip ospf 100 area 0
rtrX(config-if)# end
```

- We save the configuration:

```
rtrX# write
```

3. Checking OSPF neighbors

- The following command will show you if your router has found "neighbor"
  OSPF routers:

```
rtrX# show ip ospf neighbor

Neighbor ID   Pri   State           Dead Time   Address         Interface
10.10.0.254     1   FULL/DR         00:00:33    10.10.0.254     FastEthernet0/0
10.10.1.254     1   FULL/BDR        00:00:38    10.10.0.1       FastEthernet0/0
10.10.2.254     1   2WAY/DROTHER    00:00:35    10.10.0.2       FastEthernet0/0
```

... As the other groups are enabling OSPF on their routers, you should
see more and more neighbors appearing.

- Repeat the above command ('show ip ospf neighbor') several times, to
  watch as other routers "appear".

4. Check OSPF routes

Question:

- Which routes are we learning from OSPF ?

To find out, use the 'show ip route' command

```
rtrX# show ip route
rtrX# show ip route ospf
```

The above commands will show you all IP routes the router knows about, and the OSPF routes in particular.

Question:

- What do you notice ?

- Are there any OSPF routes (marked 'O') ?

5. Remove the static routes

Since we still have static routes, now is the time to remove them.

Do a "show ip route" on your router to see which routes you have configured:

```
...
S       10.10.6.0 [1/0] via 10.10.0.6
S       10.10.7.0 [1/0] via 10.10.0.7
...
```

For each route you see, remove it using the "no ip route command", like this:

```
rtrX# config terminal
rtrX(config)# no ip route 10.10.6.0 255.255.255.0 10.10.0.6
rtrX(config)# no ip route 10.10.7.0 255.255.255.0 10.10.0.7
...
rtrX(config)# exit
```

You should now see OSPF routes!

```
...
O>* 10.10.3.0/24 [110/11] via 10.10.0.4, eth0, 02:04:06
O>* 10.10.4.0/24 [110/11] via 10.10.0.4, eth0, 01:57:06
...
```

Question:

- For each of the groups you see routes for, can you reach the PCs in these groups with ping ?

  (For example, if you see a route for 10.10.3.0, try and ping pc9, 10, 11 or 12 (in group 3). Does it work ? Of course, choose another group if you are in group3 yourself!)

Note: it should work since you have configured OSPF, and your OSPF router sees the route for 10.10.3.0/24

If it works, it is time to save the configuration

```
rtrX(config)# write memory
```

6. Check the routes again using "show ip route" - what do you see ?

Now, use traceroute to check the path being followed!

Make sure the PC you are running traceroute to belongs to a group for
which there is an OSPF route in your router's routing table
(Hint: use "show ip route" on the router to verify)

```
% traceroute -n pc5.ws.nsrc.org

 1  10.10.X.254  1.987 ms  4.224 ms  6.456 ms
 2  10.10.0.2  8.668 ms  11.033 ms  13.065 ms
 3  10.10.2.5  15.201 ms  17.323 ms  19.478 ms
```

Note: on the GW we've done the same work already, and enabled OSPF
on it, too, so it's listening on the backbone interface, and it can see
all of your networks as you enable OSPF - tell the instructor when you
are sure it functions, so he can check.

7. Remove your router's default gateway!

We also told our GW to send out the route information for the default gateway.

It means you can now remove the default route on your cisco:

```
rtrX#
rtrX(config)# no ip route 0.0.0.0 0.0.0.0
rtrX(config)# exit
rtrX#
```

Try to check that the Internet is still reachable. Do this from your PC:

```
$ ping google.com
$ ping 8.8.8.8
```

Question:

- Does it function ? What does traceroute -n show for these destinations ?