Exercise: OSPF

Prerequisites

This exercise assumes a mostly flat topology that works as follows:

- Each router (7200 esque) with a client PC behind it and connected to every other router in the group.
    - GigabitEthernet Interface `gi 0/0` is connected to the client pc
    - GigabitEthernet interface `gi x/0` is connected to a corresponding router. There may be more than one interface.

Before doing this exercise you need to have done the following:

1. Downloaded the address plan linked to together with this exercise.
2. Out of the `/28` IPv4 address space listed for you, subnet your address space such that you have:
    - One `/29` for the link between you and your PC
    - One `/32` to be used as your router's loopback interface.
3. Out of the `/60` IPv6 address space listed for you, subnet your address space such that you have:
    - One `/64` to be used for Loopbacks (your router only needs one /128)
    - One `/64` to be used for the link between your client PC and your router.
    - *We recommend using the first /64 for the loopbacks and the second for the link to your router and using the first IP address in each /64 for the router. So for example if we have `2001:43f8:220:ffd0::/60` these will be:*
        - *`2001:43f8:220:ffd0::/64` for loopbacks*
            - *`2001:43f8:220:ffd0::1/128` for the router*
        - *`2001:43f8:220:ffd1::/64` for the point-to-point between the PC and the router*

Summary of OSPF commands

router ospf 1

Enter OSPF configuration mode. The **1** is simply the OSPF process ID - a router can run more than one OSPF process if required. We tend to recommend that people use their ASN as their process ID but it really doesn't matter. Note that process IDs do **not** even have to match between routers in the same area.

network *n.n.n.n w.w.w.w* area 0

The network statement has two functions and its use varies depending on the function required.

1. The first use is to enable OSPF on all interfaces which match the given IP address + "wildmask". A wildmask is used in access control lists to select a range of addresses. A "0" bit in the wildmask means the corresponding address bit must match, and a "1" means the corresponding address bit is "don't care". Examples:

```
network 196.200.220.17 0.0.0.0 area 0
network 196.200.220.98 0.0.0.0 area 0
   -- talk OSPF only on those interfaces with these addresses

network 196.200.220.192 0.0.0.3 area 0
   -- talk OSPF on the serial interface with this address (this is an
      alternative to the previous examples and has the same functionality)

network 196.200.220.0 0.0.0.255 area 0
   -- talk OSPF on every interface whose IP address is 196.200.220.X
```

```
network 0.0.0.0 255.255.255.255 area 0
   -- talk OSPF on every interface we have
```

By talking OSPF on an interface the router will automatically inject the network block used on that interface into OSPF.

Note that it is important that we only talk OSPF to our own networks, never to customers or other ISPs (they could break our network by injecting bad information), so usually it is best just to list the interfaces we want to talk on.

2. The second use of the network statement is to inject prefixes into OSPF. For this function the network statement must match both the network and the network's mask on that interface. The mask is called an "inverse mask", and is the one's complement of the network mask for the network block. This technique is used to inject prefixes from non-OSPF speaking interfaces into OSPF.

```
network 196.200.220.192 0.0.0.3 area 0
   -- announce the network 196.200.220.192/30 to OSPF

network 196.200.220.0 0.0.0.255 area 0
   -- announce the network 196.200.220.0/24 to OSPF
```

redistribute connected subnets

Advertise all networks to which we are connected, including those which are not being used to talk to other OSPF routers. The use of this command is not recommended within ISP backbones, but is included here for completeness.

("redistributed connected" by itself only distributes classful routes, i.e. whole class A/B/C networks, so it's important to add "subnets")

redistribute static subnets

Advertise all static routes we have (except defaultroute). The use of this command isn't ordinarily required or recommended. However, there will be circumstances where redistribution of statics into OSPF are required. An example will be shown in the lab.

default-information originate [always] metric *n*

Advertise a default route into OSPF, with a cost of "n". Typically this would go on your border router(s). Without 'always', the announcement will only be made if the router already has a valid defaultroute from somewhere else (e.g. a static route to a link which is up)

area 0 authentication message-digest

Use MD5 authentication on all OSPF packets

Per-interface configuration

```
ip ospf cost <n>                           Set outbound interface cost
ip ospf message-digest-key 1 md5 <string>   Set MD5 authentication key
```

Part 1: Loopback interfaces

A loopback interface is a single (/32) IP address which belongs to a device, independent of its physical interface addresses. It's very convenient when managing routers, because you can use the loopback address as a fixed address to telnet to, or monitor using SNMP, which will continue to work even if one or more of the interfaces has failed.

1. Check that there is no OSPF process running from any previous exercises. If it is still there, remove it.

   ```
   router-a#conf t
   router-a(config)#no router ospf 1
   router-a(config-if)#  [Hit ctrl-Z]
   router-a#write mem
   ```

2. Check that you have no static routes. If you do have some, remove them.

   ```
   router-a#show ip route
   The only routes you should see are (C)onnected routes for your own
   interfaces. If you have any (S)tatic routes, delete them like this:
   router-a#conf t
   router-a(config)#no ip route 196.200.220.32 255.255.255.240 196.200.220.30
   router-a(config-if)#  [Hit ctrl-Z]
   router-a#write mem
   ```

3. The router is now ready for this exercise.

   ```
   router-a#reload
   ```

4. Log in and check your router can still ping the PC on your desk, and other routers on the backbone.

   (Your PC won't be able to ping any other PCs though, because your router doesn't have the routes any more)

5. As described in the presentation the loopback interface address is used to form OSPF's router-id. If no loopback is configured, the router uses the highest IP address configured at the time the OSPF process was started.

   You will need to subdivide your assigned network space. You will need one subnet for your desk network, and a /32 for the loopback. (In real life you would have a number of routers, and say a /29 block would be enough for 8 loopbacks). Configure your loopback address into your router.

   The example below is for the router belonging to Team A. They have decided to subdivide their address block 196.200.220.32/28 such that 196.200.220.47/32 is the address of the loopback interface of their router.

   ```
   router-a#conf t
   router-a(config)#int loopback0
   router-a(config-if)#ip address 196.200.220.47 255.255.255.255
   router-a(config-if)#  [Hit ctrl-Z]
   ```

6. Now save your configuration.

   ```
   router-a#write mem
   ```

Part 2: Learning routes via OSPF

The network topology is the same as for the static routing exercise

1. Enable OSPF on the backbone interface (only).

We will use a feature introduced in IOS 12.0 which explicitly disables OSPF on all interfaces except those you nominate.

```
    router-a#conf t
    router-a(config)#router ospf 1
    router-a(config-router)#log-adjacency-changes
    router-a(config-router)#passive-interface default
    router-a(config-router)#no passive-interface gi1/0
    router-a(config-router)#network 196.200.220.0 0.0.0.31 area 0 ! backbone
    router-a(config-router)#network 196.200.220.47 0.0.0.0 area 0 ! loopback
    router-a(config-router)#network 196.200.220.32 0.0.0.7 area 0 ! gi0/0subnet
    router-a(config-router)#  [Hit ctrl-Z]
```

2. We will use MD5 authentication, to ensure that we only listen to OSPF packets from machines which know the secret key; we will also set the link cost to 100.

```
    router-a(config)#router ospf 1
    router-a(config-router)#area 0 authentication message-digest
    router-a(config-router)#int gi1/0
    router-a(config-if)#ip ospf message-digest-key 1 md5 afnog
    router-a(config-if)#ip ospf cost 100
    router-a(config-if)#  [Hit ctrl-Z]
```

In real life you should use an MD5 key which is different from your login, enable and SNMP strings

3. Look at OSPF status

```
    router-a#show ip ospf int
    router-a#show ip ospf neighbor
```

To interpret the neighbor information:

```
2WAY    =  we are neighbors (we have established 2-way exchange of hellos),
           but neither of us is a designated router
FULL    =  we are neighbors and we exchange routes (one of us is DR or BDR)

DR      =  we are the Designated Router for this network
BDR     =  we are the Backup Designated Router for this network
DROTHER =  we are neither DR nor BDR
```

If you see other states, they are intermediate steps on the way to establishing the final relationship, and should change after a few seconds.

4. Question: who is the Designated Router (DR) and Backup Designated Router (BDR) on the backbone network? **Write down your answers bellow**.

```
DR =                           BDR =
```

5. Once you have established a neighbor relationship with another router, you should automatically have learned some new routes:

```
router-a#show ip route
```

Routes learned through OSPF are tagged with **O**. Check that the *next hop IP address* for each route is correct

Also, the *far* router should also have picked up *your* route. You can go over to the other desk and ask to see "show ip route"

6. Check that your PC can ping the PCs on other desks

   ```
   $ ping 196.200.220.129
   ...
   ```

Part 3: IPv6 OSPF

We are now going to take the above topology and enable IPv6 on it.

1. To enable OSPF for IPv6 (OSPFv3), we do:

   ```
   ipv6 router ospf 1
   ```

   We also need to mark all interfaces passive by default, as we did for OSPFv2 (for IPv4 above), and then activate the ones we will talk OSPFv3 over. For example:

   ```
   ipv6 router ospf 1
    log-adjacency-changes
    passive-interface default
    no passive-interface gi1/0
   ```

2. We now add an IPv6 address for the interfaces we will be running IPv6 over. The active interface will be the classroom backbone, so the example might be:

   ```
   interface gigabitethernet 1/0
    ipv6 address 2001:43f8:220:ff00::1/64
   ```

   Configure addresses on the active interfaces on the router. You will need one /64 for the link to your PC, one /128 for the loopback interface, and of course an address on the backbone network.

3. Rather than using network statements as in IPv4, we simply go to each interface we want to run OSPFv3 on, and activate it for IPv6. Here is an example:

   ```
   interface gigabitethernet 0/0
    ipv6 router ospf 1 area 0
   ```

   So this example says that we are going to run OSPFv3 on GigabitEthernet 0/0, and that this interface will be in area 0. It also says that the IPv6 address/subnet configured on GigabitEthernet 0/0 will be automatically inserted into the OSPFv3 routing table.

   Interfaces which are marked as passive in the generic OSPFv3 configuration but have the

   ```
   ipv6 router ospf 1 area0
   ```

   marked on the interface will simply have their configured IPv6 network address being inserted into the OSPFv3 routing table.

Last updated ndg&fh, 2017-05-22