

OSPFv2 (Part 6)

OSPF Optimization and Troubleshooting

Agenda

- Pop quiz
- New material:
 - OSPF optimization and troubleshooting
 - General troubleshooting strategies – ENSA module 12

Assignments and Lab work

- Lab 5: Multi-area OSPFv2, this week
- Lab 5 post-lab: due **before** your lab session next week
- Lab 6: a detailed study of ping, traceroute, and DNS
- Readings:
 - Module 12 of the ENSA course on NetAcad, completed by next Wed.
 - Supplementary slide deck for Troubleshooting

The job of any routing protocol, including OSPF, is to find a route to every (reachable) subnet. All routes from a protocol are candidates for the routing table.

SPF / How OSPF chooses the *best* route

- Based on accumulated cost (simple, basic addition of cost of each link in path)
- In order to do that, it must know the cost of each link (or cost to get to a subnet)

Optimizing OSPF

- SPF calculation is actually meaningful: costs must be accurate – *auto-cost reference-bandwidth {value}* Consider this **mandatory!**
- Minimizing delays upon power on: unless DR/BDR elections are truly necessary, eliminate them to remove their ~1/2 minute delay
ip ospf network point-to-point
- Passive interfaces: reduce the 'spam' sent to hosts that don't/aren't running OSPF *passive-interface {i/f name}*
- Hello / Dead timers: looking for the Goldilocks zone: a balance between extra processing for the routers' CPUs and having the shortest possible recovery time if/when a router fails. For 'smaller' networks, can reduce from the default. (For those who insist on knowing, try: **show processes cpu**)

Troubleshooting OSPF

1

2

Two main categories of problems with OSPF: not connected, and missing some info. You'll need to know *what show commands* provide confirmation of each item.

- 1 – Router ID: double-check that it's unique and correct (multiple "show..." commands display this because it's so important!)

Critical importance of the info in the interface, neighbour, database tables

- 1 – Missing interfaces: I suggest easier to debug missing interfaces with interface config method (*show ip ospf inter br* and *show ip proto*)
- 1 – Areas: cannot directly connect different areas unless one of them is backbone! (results in neighbours missing from neighbour table)
- 1 – Missing neighbours:
 - Matching requirements for both ends of a link: (unique router ID as above), subnet ID, mask len, area #, hello & dead timers, link-type (Bcast vs P2P) [plus authentication, area-type - NET3008]
Ref: Wk03-OSPFv2Basics, p. 17; Wk03-OSPFv2Config, p. 60
 - Interface must not be passive
- 2 – Routing table, overall status details in *show ip proto* (e.g ABR?, ASBR?)
- 2 – Missing routes: are all required interfaces included in OSPF? (Think: how is this different from missing interfaces in the interface table?)
- 2 – Interface down, so router not participating in that area, so maybe no longer an ABR, and thus receiving none of the routes from adjacent area.

Even at our stage of learning OSPF, Packet Tracer just isn't adequate:

```
R1#show ip ospf interface brief
      ^
% Invalid input detected at '^' marker.
R1#show ip ospf interface

GigabitEthernet0/0 is up, line protocol is up
Internet address is 10.1.2.1/24, Area 1
[... lots of additional info follows ...]
```

Though the info from PT contains all the details, we're forced to wade through lines and lines of info, instead of having the nice, brief table of info (1 line per interface).