Post 16

Thursday, January 24, 2019 11:15 PM

DYNAMIC ROUTING P3

Why Use Link-State Protocols?
Advantages:

Builds Map	Each router builds own topological map of network to determine shortest path
Fast Convergence	Immediate flooding of LSP's achieves faster convergence • RIP needs to process each update/update table before flooding out
Event-driven Updates	LSP's only sent when change in topology/contain only what's needed • Don't send periodic updates
Hierarchical	Use of areas: Multiple areas create hierarchical design to networks • Better route summarization/isolation of r-issues w/in area

Disadvantages:

Mem	Additional mem to create/maintain db/SPF tree
Processing	More CPU power over distance vector • SPF = more CPU time over distance vector = Bellman-Ford B/c build complete map of topology
BW	Flooding of LSP's can affect BW

Hierarchical Design by Areas: Modern link-state r-protocols designed to min effects on mem/CPU/BW

- Use/config of multiple areas can reduce size of link-state db's
- Multiple areas can limit amt of info flooding in r-domain/send LSPs only to needed
- Change in topology? Only routers in affected area receive LSP/run SPF alg
- Can help isolate unstable link

Protocols that Use Link-State: Only 2 link-state routing protocols: OSPF/IS-IS

OSPF	• Designed: IETF OSPF Working Group: 1987 • 2 versions in use: 1. OSPFv2: IPv4 [RFC 1247/RFC 2328] 2. OSPFv3: IPv6 [RFC 2740]
IS-IS	• Designed: ISO 10589: 1st dev at DEC: Phase V. Radia Perlman: Chief designer • Originally OSI protocol suite: NOT TCP/IP • Later: Dual IS-IS included support for IP

R-Table Entries: Hierarchy in IOS originally implemented w/classful scheme

- Incorporates both classful/classless: Structure still built around classful

Directly Connected Entries:
Entries contain the following information:

Route source	ID's how route learned Directly connected ints have 2 route source codes: **1. C**: ID's directly connected network: Auto created when int config w/IP/activated **2. L**: ID's local route: Auto created when int config w/IP/activated
Destination network	Address of remote network/how network is connected
Outgoing int	ID exit int to use when fwding packets to destination network

Router multiple ints config'd: Table stores info about directly connected/remote routes

S	Manually created: AKA static route
D	Learned dynamically from another router using EIGRP
O	Learned dynamically from another router using OSPF

R	Learned dynamically from another router using RIP

Remote Network Entries:

Route source	How route learned
Destination network	Address remote network
AD	Trustworthiness route source
Metric	Value assigned to reach remote network: Lower preferred
Next hop	IPv4 address of next router to fwd packet
Route timestamp	When route last heard
Outgoing int	Exit int to use to fwd packet toward final dest

R-Table Terms: Table hierarchical: **Routes discussed in terms of:**

1. Ultimate route
2. Level 1 route
3. Level 1 parent route
4. Level 2 child routes

Ultimate	R-table entry contains either next-hop IPv4/exit int • Directly connected | Dynamically learned | Local routes
Level 1	Route w/subnet mask = or less than classful mask of network address **Can be a:** • **Network:** Route that has subnet mask = classful mask • **Supernet:** Network address w/mask less than classful mask [summary address] • **Default:** Route is static w/address 0.0.0.0/0 • Directly connected network | Static route | Dynamic routing protocol Can also be ultimate
Level 1 Parent	Route is subnetted: Parent route can never be ultimate
Level 2 Child	Route that is subnet of classful network address Can be a: • Directly connected network | Static route | Dynamically learned route • Also ultimate routes

Route Lookup Process: When packet arrives: Router examines header/ID's destination IPv4/router lookup process

1. If match lvl 1 ultimate: Route used to fwd packet
2. If match lvl 1 parent: Proceed to next step
3. If match lvl 2 child: Subnet used to fwd packet
4. If no match w/lvl 2 child: Proceed to next step
5. If lesser match w/lvl 1 supernet/default: Router uses to fwd packet
6. If no match any route in table: Packet dropped

Best Route = Longest Match: For there to be match between destination IPv4 of packet/route in table:

a. A min # of far left bits must match between IPv4 of packet/route in table
b. Subnet mask of route in table used to determine min # of far left bits
c. IPv4 packet only contains address: Not subnet mask

Route w/the greatest # of equiv far left bits: AKA longest match: ALWAYS preferred

IPv6 Routing Table Entries: Similar to IPv4: Uses directly connected ints/static/dynamically learned routes

- IPv6 classless: All routes lvl 1 ultimate: No lvl 1 parent of lvl 2 child routes

Directly Connected Entries

show ipv6 route

Directly connected route entries display:

Route source	How route learned: 2 source codes: **1. C**: Directly connected **2. L**: Local
Directly connected	IPv6 address of
AD	Trustworthiness of source: IPv6 uses same distances as IPv4

Remote IPv6 Network Entries

Route source	How route learned: O: OSPF, D: EIGRP, R: RIP, S: Static

When IPv6 packet arrives on router int: Router examines IPv6 header/ID's destination IPv6:

- Router goes through router lookup process: Examines lvl 1 routes for best match w/destination address of packet
- Longest match = best match