Post 15

Thursday, January 24, 2019 11:15 PM

DYNAMIC ROUTING P2 EIGRP/RIP/OSPF

RIP: Routing Information Protocol: 1st gen IPv4: Specified originally in RFC 1058: Easy config: Good for small networks

Characteristics:

- Updates broadcasted every 30 seconds: 255.255.255.255
- **Hop count used as metric: Greater than 15 hops: Deemed infinite [too far]**
  - 15th hop router wouldn't propagate r-update to next router

RIPv2 improvements: 1993: Updated to classless: **1997 IPv6 version released:** Still has 15 hop limit/AD of 120

Classless	VLSM/CIDR: Includes subnet mask in r-updates
More Efficiency	Fwds updates to multicast 224.0.0.9 instead of broadcast 255.255.255.255
Reduced Entries	Manual route summarization on any int
Secure	Authentication mechanism to secure table updates between neighbors

Updates encapsulated in UDP segment: Both source/destination ports set to UDP 520

EIGRP: Enhanced Interior-Gateway Routing Protocol

IGRP: Interior Gateway Routing Protocol: 1st proprietary IPv4 developed by Cisco: 1984

Characteristics:

- **BW | Delay | Load | Reliability | Used to create composite metric**
- R-updates broadcast every 90 seconds

EIGRP: 1992 IGRP was replaced: Also introduced VLSM/CIDR

- **Increased efficiency | Reduces r-updates | Supports secure msg exchange**

EIGRP introduced:

Bounded triggered updates	○ NO periodic updates ○ Table changes propagated whenever change occurs ○ Reduces load on network ○ EIGRP only sends to neighbors that need it
Hello keepalive mechanism	○ Hello msg periodically exchanged to maintain adjacencies w/neighbors ○ Very low resource usage
Maintains topology table	○ Maintains all routes received from neighbors in table (not just best paths) ○ DUAL can insert backup routes
Rapid convergence	○ Fastest IGP to converge: Maintains alternate routes: Enables fast convergence ○ If primary route fails: Router can use alternate route ○ Switchover to alternate doesn't involve interaction w/other routers
Multiple network layer protocol support	○ EIGRP uses PDM: Protocol Dependent Modules ○ Meaning: Only protocol to include support for protocols other than IPv4/IPv6 ▪ Example: Legacy IPX/AppleTalk

RIP Config Mode

router rip [*global config*] Doesn't directly start RIP: Provides access to router config mode where settings are config'd

no router rip [*global config*] Disable: Stops RIP process/erases all existing RIP configs

address-family	Address family cmd mode
auto-summary	Enable auto network # summarization
default	Set a cmd to default
default-information	Control distribution of default info
default-metric	Set metric of redistributed routes
distance	Define an AD
distribute-list	Filter networks in r-updates
exit	Exit r-protocol
flash-update-threshold	Specify flash update threshold in seconds
input-queue	Specify input queue depth
maximum-paths	Fwd packets over multiple paths
neighbor	Specify a neighbor router
network	Enable routing on an IP network
no	Negate a cmd/set its defaults
offset-list	Add/subtract offset from RIP metrics
output-delay	Interpacket delay for RIP updates
passive-interface	Suppress r-updates on an int
redistribute	Redistribute info from another r-protocol
timers	Adjust r-timers
validate-update-source	Perform sanity checks against source address of r-updates
version	Set r-protocol version

Advertising Networks
RIP config: Starts RIP: Needs local ints to use w/other routers/which locally connected networks to advertise to them

network *network-address* [router config] Enable RIP routing for a network
- Enter classful network address for each directly connected network
- Enables RIP on all ints that belong to a specific network: Associated ints now send/receive RIP updates
- Advertises specified network in RIP updates sent to other routers every 30 seconds
- If subnet address entered: IOS auto converts it to classful network address

show ip protocols Displays IPv4 r-protocol settings currently config'd
show ip route Displays RIP routes installed in table

Enabling RIPv2: Default: When RIP is config'd on Cisco: It runs RIPv1: RIPv1 ignores RIPv2 fields in route entries
- Even though the router only sends RIPv1 msgs: It can interpret RIPv1/RIPv2 msgs

version 2 [*global config*] Enables RIPv2
show ip protocols Verifies R2 is config'd to send/receive version 2 msgs only
- RIP process includes subnet mask in all updates, making RIPv2 classless
- The version 2 cmd must be config on all routers in r-domain

Disabling Auto Summarization: RIPv1/RIPv2 auto summarizes networks at major network boundaries by default
no auto-summary [*router config*] RIPv2 no longer summarizes networks to their classful address at boundary routers
- No effect w/v1: With v2, it includes all subnets/appropriate masks in updates

Passive Interfaces: Default: RIP updates fwded out all RIP enabled ints

- Updates only need to be sent out of ints connecting other RIP enabled routers

Unneeded update impact on LANs:

Wasted BW	BW used to transport unnecessary updates • RIP either broadcasted/multicasted: Switches also fwd updates out all ports
Wasted Resources	All devices on LAN must process update up to transport: Then discard
Security Risk	○ Advertising updates on broadcast network is BAD: Can be intercepted w/packet sniffing tools ○ Updates can be modded/sent back to router: ▪ They can corrupt the tables w/false metrics that misdirect traffic

passive-interface [*router config*] Prevents transmission of r-updates through a router int
- Still allows that network to be advertised to other routers
- The network that the specified int belongs to: Still advertised in updates sent out other ints
- All r-protocols support passive-interface cmd

passive-interface default All ints can be made passive
no passive-interface Can re-enable ints that don't need to be passive

Propagating Default Route: Edge router must be config with:
ip route 0.0.0.0 0.0.0.0 *exit-intf next-hop-ip* Default static route
default-information originate Tells router to originate default info, by propagating static default route in RIP updates

Advertising IPv6:
ipv6 unicast-routing Enable router to fwd IPv6 packets: Must be config
Instead of network network-address use: ipv6 rip *domain-name* enable
Propagating a route in RIPng is identical to RIPv2 except an IPv6 default static route must be specified.

To propagate must be config with:
ipv6 route 0::/0 2001:DB8:FEED:1::1 [global config] A default static route
ipv6 rip domain-name default-information originate Instructs X to be source of default route info
- Propagates default static route in RIPng: Updates sent out of config int

RIPng Config:
show ipv6 protocols Doesn't provide same amt of info as IPv4 counterpart
- **It does confirm**: 1. RIPng routing is config/running 2. The int config w/RIPng.

show ipv6 route Displays routes installed in table
- RIPng: The sending router already considers itself to be 1 hop away

Shortest Path First Protocols: AKA Link-State Routing Protocols
- Built around Edsger Dijkstra's SPF: Shortest Path First algorithm

IPv4 link-state r-protocols: OSPF: Open Shortest Path First | IS-IS: Intermediate System-to-Intermediate System
- Have reputation of being much more complex than distance vector counterparts
- Basic function/config straight-forward

OSPF operations can be config using:
router ospf process-id [global config]
network To advertise networks

Dijkstra's Algorithm AKA SPF: All link-states apply this alg to calc best path route
- Uses accumulated costs along each path from source/destination: Determines total cost of route

Link-State R-Process: How does it work?
Link-state: Info about state of links

Link-state routing process	• **Meeting neighbors on directly connected networks:** Exchange Hello packets 1. Other link-state routers on directly connected networks • **Build LSP: Link-State Packet:** Contains state of each directly connected link 1. Records info about neighbors [ID/link type/BW] • **Floods LSP to all neighbors:** Neighbors store LSPs in db 1. Flood LSPs until all routers in area receive

	2. Stores copy of each LSP from neighbors in local db • **Uses db to construct map/best path to each destination** 1. SPF alg construct map of topology/determines best path 2. Process same for OSPF [IPv4/IPv6]

Link & Link-State Int must be included in 1 of r-config statements before participating in link-state r-process

Say Hello: 2nd step in link process: Each router responsible for meeting neighbors

- Routers w/link-state use Hello protocol to discover neighbors on links

Neighbor: Any other router enabled w/same link-state r-protocol

Adjacency: When 2 link-state routers learn they are neighbors

- Hello packets continue exchanges between 2 adjacent neighbors: Keepalive function monitors
- If router stops receiving: Neighbor considered unreachable/adjacency broken

Building the Link-State Packet: 3rd step in process: Each router builds LSP containing state of directly connected links

- After router establishes adjacencies: Can build its LSPs containing link-state info about links

Flooding the LSP: 4th step in process: Each router floods LSP to all neighbors: They store in db

- When router receives LSP from neighbor: It sends that LSP out all ints except 1 receiving LSP
- Process creates flooding effect
- Link-state r-protocols calc SPF alg after flooding complete
- As result: Reach convergence quickly

LSPs don't need to be sent periodically: An LSP only needs to be sent:

1. During initial startup on router

Example: Restart

1. When a change in topology

Example: Link going down/coming up/neighbor adjacency being established/broken

1. Addition to l-s info: Other info in LSP: Sequence #s/aging to help manage flooding
    1. Info used by router to determine if received LSP from other router/LSP has newer info
    2. Process allows router to keep most current info in db

Building Link-State Db: Final step: Each router uses db to construct a map of topology/computes best path

- LSPs are stored in link-state db

Building SPF Tree	○ Each router in r-area uses link-state db/SPF alg to construct SPF tree ▪ SPF alg interprets each router's LSP to ID networks/costs ▪ SPF calcs shortest paths to reach each individual network: Making SPF tree ▪ Each router constructs its own independent SPF tree ○ To ensure proper routing: link-state db's used to construct trees must be identical on all routers

Adding OSPF Routes to Table: Using shortest path info by SPF alg, paths added to table

- Table includes all directly connected networks/routes from other sources (like static routes)
- Packets now fwded according to entries in r-table