EIGRP

Lab

EIGRP - Introduction

- Enhanced Interior Gateway Routing Protocol
- IGRP Evolution
- Advanced distance vector protocol
- Cisco proprietary
- DUAL algorithm (Diffused Upade Algorithm)
- Multicast address: 224.0.0.10
- Administrative distance: 90

EIGRP - Introduction

- Automatic neighbor discovery with **hello packets**
- Adjacency mechanism (same subnet, same K-values, same AS, same authentication)
- **Update packets** used to exchange routing information (full / partial)
- Topology exchanged the first time, and then triggered partial updates
- Analyses its « topology » table and then selects the best path
- Uses RTP (Reliable Transport Protocol) as transport protocol
- Supports IP, IPv6, IPX, AppleTalk

EIGRP - Introduction

- Complex metric formula
- Default parameters: bandwidth and delay
- Other parameters: load, reliability, MTU (but MTU is not taken into account in metric calculation)
- Best route = lowest metric
- Only best routes are in the routing table

EIGRP - Introduction

- FD (Feasible Distance) = metric of the best route to the destination
- RD (Reported Distance) = metric of a route to the destination received from a neighbor (received in update packets)

- Successor = best route to the destination (installed in RT)
- FS (Feasible Successor) = optionnal, second best route to the same destination. EIGRP can switch over in case of successor route failure.

- If no FS, EIGRP initiate query/reply process to find a new route
- That is why EIGRP is considered as « active »
- SIA problem discussed later

EIGRP - Introduction

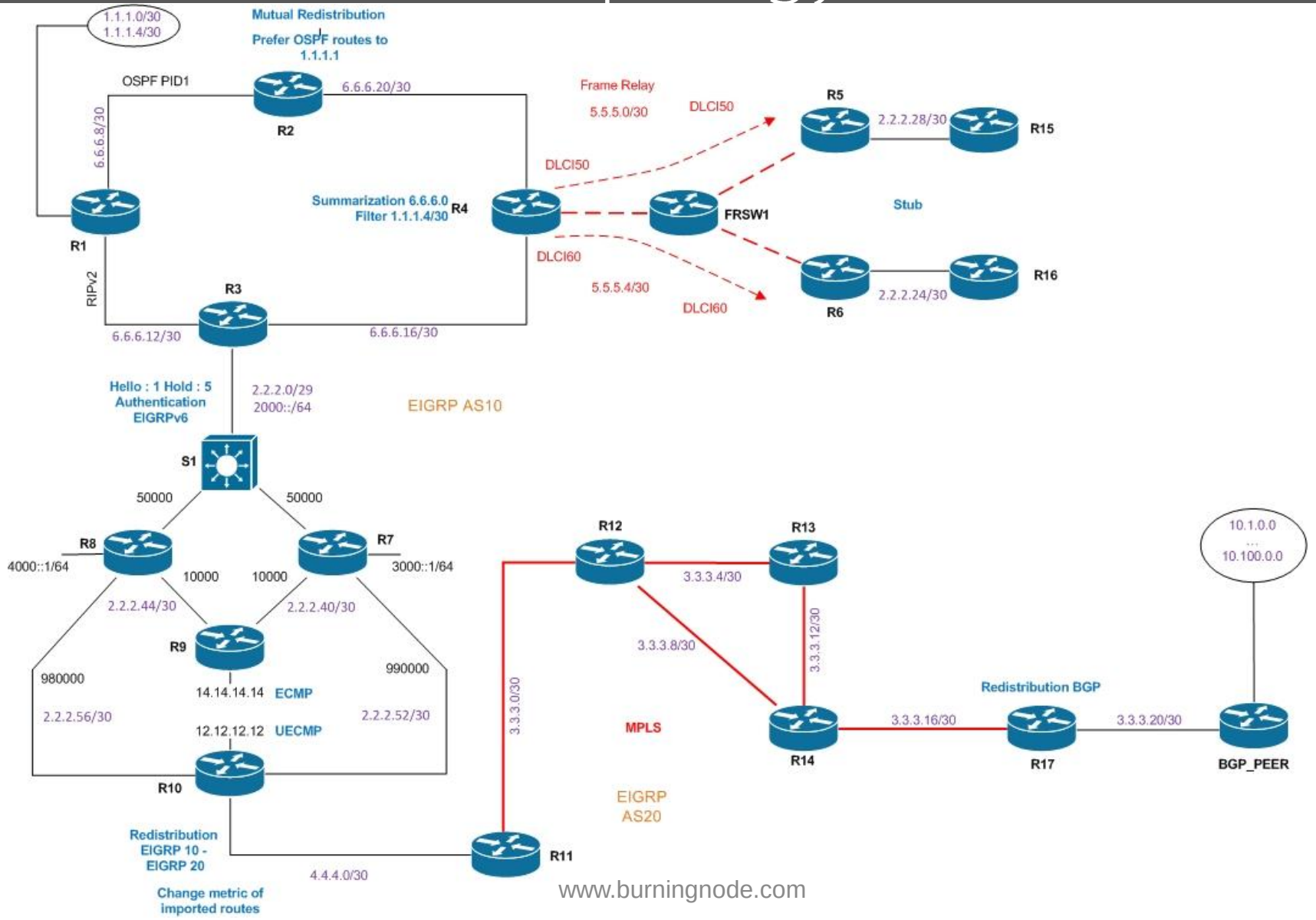
- EIGRP calculates a metric for all routes, and selects the best (Successor) to put in the routing table

- Then it looks for a FS regarding the **Feasibiliy Condition (FC)**

- For a non-successor route R, if RD< FD
- Then R can be FS
- Else it is not considered FS (loop avoidance)

- A word on EIGRP design :
  - Single area design, (flat)
  - two-level hierarchical design using stub and summarization features

Topology


1.1.1.0/30
1.1.1.4/30
Mutual Redistribution
Prefer OSPF routes to 1.1.1.1
OSPF PID1
6.6.6.20/30
R2
6.6.6.8/30
R1
Summarization 6.6.6.0
Filter 1.1.1.4/30
R4
Frame Relay
5.5.5.0/30
DLCI50
DLCI50
R5
2.2.2.28/30
R15
FRSW1
Stub
DLCI60
5.5.5.4/30
DLCI60
R6
2.2.2.24/30
R16
RIPv2
R3
6.6.6.12/30
6.6.6.16/30
Hello : 1 Hold : 5
Authentication
EIGRPv6
2.2.2.0/29
2000::/64
EIGRP AS10
S1
50000
50000
R8
4000::1/64
R7
3000::1/64
10000
10000
2.2.2.44/30
2.2.2.40/30
R9
980000
990000
14.14.14.14 ECMP
2.2.2.56/30
2.2.2.52/30
12.12.12.12 UECMP
R10
Redistribution
EIGRP 10 -
EIGRP 20
Change metric of
imported routes
4.4.4.0/30
R11
R12
R13
3.3.3.4/30
3.3.3.8/30
3.3.3.12/30
3.3.3.0/30
MPLS
R14
3.3.3.16/30
Redistribution BGP
R17
3.3.3.20/30
BGP_PEER
10.1.0.0
...
10.100.0.0
EIGRP
AS20

Tasks

- Implement mutual redistribution between RIPv2, OSPF and EIGRP, prefer OSPF routes and filter prefix 1.1.1.4/32 from being learnt
- Implement EIGRP over Frame Relay (NBMA) / Summary
- Implement query scoping (stub networks) on branch offices
- Tune EIGRP timers,metrics and add authentication
- Comment EIGRP ECMP
- Implement EIGRP for IPv6
- Implement EIGRP NECMP
- Achieve mutual EIGRP – EIGRP redistribution, tune metrics of imported routes and change AD values
- Implement EIGRP as underlying MPLS LDP routing protocol and use load in the metric calculation for this network (and prevent eigrp from advertising on all intf)
- Redistribution of BGP routes inside EIGRP AS
- Discuss SNMP, security, and NSF features

EIGRP – Redistribution

- Route tagging to prevent recursive learning between OSPF and RIP through EIGRP

✓ Route-map *redistribution* to apply the tags and route-map *checkin* to deny tag 11 and 22 (OSPF & RIP)

- Modify metric of OSPF learnt routes

✓ In the route-map *redistribution* change the values of ospf routes to modify EIGRP metric calculation

- Simple mutual redistribution

✓ `#redistribute eigrp 10 metric 13 route-map CHECK-TAG (rip)`

✓ `#redistribute eigrp 10 metric-type 2 subnets route-map CHECK-TAG (ospf)`

✓ `#redistribute rip metric 100000 100 255 1 1500 route-map redistribution`

✓ `#redistribute ospf 1 metric 100000 100 255 1 1500 route-map redistribution`

EIGRP – Filtering incoming prefixes

- Create access-list to match the 1.1.1.4/32 prefix

✓ `#ip access-list standard prefix_to_filter`

✓ `#permit 1.1.1.4 0.0.0.3`

- Filter it under EIGRP process

✓ `#distribute-list prefix_to_filter in`

EIGRP over FR (NBMA) / Summary

- Create a summary route for the 6.6.6.0 network on R4 (serial if)

✓ `ip summary-address eigrp 10 6.6.6.0 255.255.255.0 15`

- Leak- maps

- Setup EIGRP neighbors to use unicast request instead of multicast

✓ network + neighbor statements

- Another solution: use FR pseudo broadcast features

EIGRP – Query scoping

- Different methods to achieve this one :
  - Stub
  - Different AS
  - Network summarization
  - Route filtering

- Here the stub feature is a good idea because the branch offices do not have any routers behind (actually the routers behind R5 and R6 simulate hosts)

- Activate stub feature on R5 and R6

✓ `#eigrp stub connected`

✓ `#sh ip eigrp neighbors detail`

EIGRP – Timers & metric tuning

- Timers tuning :

✓ `#ip hello-interval eigrp 10 2`

✓ `#ip hold-time eigrp 10 8`

- Metric calculation tuning :
- 256*[(K1*Bw) + (K2*Bw)/(256-Load) + K3*Delay)*(K5/(Reliability + K4))]
- K1 = Bandwidth K2 = Load K3 = Delay K4 = Reliability K5 = MTU
- Default : #metric weights 0 1 0 1 0 0

✓ `#metric weights 0 1 0 0 0 0`

- Key values must be the same to form an adjacency, it is advised to also keep consistent timer across the network

EIGRP – Authentication

- Create key chain to store the password

✓ `#key chain EIGRP`
✓ `(config-keychain)#key 0`
✓ `(config-keychain-key)#key-string lab`
✓ `(config-keychain-key)#accept-lifetime 00:00:00 Jan 1 1993 infinite`

- Activate authentication on the interface

✓ `#ip authentication key-chain eigrp 10 EIGRP`

- Set mode to md5 digest to hash authentication information (optionnal, only option since IOS 12.0)

✓ `#ip authentication mode eigrp 10 md5`

EIGRP – ECMP

- Setup R10 with a loopback Lo0 8.8.8.8

```
D       8.8.8.0 [90/256000] via 2.2.2.3, 00:00:12, FastEthernet0/0
                [90/256000] via 2.2.2.2, 00:00:13, FastEthernet0/0
```

- Restrict the number of equal cost paths :

```
R14(config-router)#maximum-paths ?
  <1-16>  Number of paths
```

- Load sharing across multiple equal paths

```
R14(config-router)#traffic-share ?
  balanced  Share inversely proportional to metric
  min       All traffic shared among min metric paths
```

- Useful to provide faster convergence in some cases

EIGRP for IPv6

- Activate IPv6 support

✓ `#ipv6 unicast-routing`

- Setup ip adresses and EIGRP on the interface

✓ `#ipv6 address 2000::1/64`

✓ `#ipv6 eigrp 10`

- Check EIGRP with

✓ `#show ipv6 protocols`

EIGRP - NECMP

- Unequal cost multipath : use multiple links with different metrics
- RD must still be inferior or equal to FD for the route to be considered
- Variance number indicate the maximum FD to consider : FD * variance
- On R3 to contact R10

```
#variance 5
D       14.14.14.0 [90/1280000] via 2.2.2.57, 00:00:04, FastEthernet1/0
                   [90/256000] via 2.2.2.53, 00:00:04, FastEthernet0/1
```

EIGRP to EIGRP redistribution

- Simple redistribution under EIGRP processes on R12 :

✓ `redistribute eigrp 20 metric 100000 100 255 1 1500`

✓ `redistribute eigrp 10 metric 100000 100 255 1 1500`

EIGRP – Metric/AD Manipulation

- Access-lists to match routes :
  - ✓ `Standard IP access list change`
  - ✓ `    10 permit 4.4.4.0, wildcard bits 0.0.0.3`
  - ✓ `    20 permit 10.0.0.0, wildcard bits 0.255.255.255`
  - ✓ `    30 permit 3.3.3.0, wildcard bits 0.0.0.255`

- Offset-list to change the metric :
  - ✓ `#offset-list change in 121000 fastEthernet 2/0`

- Distance to change the AD :
  - ✓ `#distance 91 171`

EIGRP - MPLS

- Traditional EIGRP network

- MPLS and LDP

✓ `#mpls ip propagate-ttl`

✓ `#mpls label protocol ldp`

- Interface config

✓ `#mpls ip`

EIGRP – BGP redistribution

- TCL to generate loopbacks

✓
```
#tclsh
```

✓
```
# set i 0 ; while {$i <100}  {ios_config "int loopback $i" "ip address
10.$i.1.1 255.255.0.0” ; incr i }
```

- Redistribute connected loopbacks in BGP

✓
```
#router bgp 65100
```

✓
```
#redistribute connected
```

✓
```
#show ip bgp neighbors 5.5.5.1 advertised-routes
```

- Select prefixes you want to learn with a route-map

- BGP redistribution inside EIGRP

✓
```
# redistribute bgp 65000 metric 100 1000 255 1 1500 route-map bgpredis
```

EIGRP – SNMP

- Can monitor device using SNMP
- Need Cisco MIBs
- EIGRP traps

✓ `# snmp-server enable traps eigrp`

- Test with Centreon or FAN

EIGRP – NSF

- NSF: Non Stop Forwarding
- SSO: Stateful SwitchOver

- Goal : maintain packet flow in the event of a router/switch failure

- SE redundancy mechanism on Cisco 4500/6500 (switches) and 7500/10000/12000 (routers)

- EIGRP is aware of NSF and the benefits are :
  - Packets continue to be forwarded during the switch over
  - EIGRP maintain the neighborship with the failed router during the switch over

- More infos

EIGRP – Security

- Authentication

- Passive Interface

✓ `#passive-interface default`

✓ `#no passive-interface X`

- If not secured, vulnerable to MiTM and DoS

- Test with Loki tool

Check

- ✓ Implement IPv4 Enhanced Interior Gateway Routing Protocol (EIGRP)
- ✓ Best path
- ✓ Loop-free paths
- ✓ EIGRP operations when alternate loop-free paths are available, and when they are not available
- ✓ EIGRP queries
- ✓ Manual summarization and autosummarization
- ✓ EIGRP stubs
- ✓ Authentication
- ✓ Composite Metric Manipulation
- ✓ Applying Offsets to Metrics
- ✓ Adjusting Timers

Check

- ✓ Unicasting updates
- ✓ Use of the 0.0.0.0 in the network command
- ✓ Manipulate the Bandwidth used by EIGRP
- ✓ Distribute lists
- ✓ Route Map Support
- ✓ SNMP Support
- ✓ Passive Interface
- ✓ NSF Awareness
- ✓ Router ID
- ✓ Implement EIGRP version 6 (EIGRPv6)
- ✓ Summarization
- ✓ EIGRP Security

Thank you