# IGRP and EIGRP

Fulvio Risso

Politecnico di Torino

# Interior Gateway Routing Protocol (IGRP)

- Distance Vector
- Cisco-proprietary
- More “marketing-oriented” features than RIP
  - More sophisticated metrics
  - Multipath
  - Less traffic (DV update every 90 seconds)
- Same problem of the missing netmask

# Why do we present IGRP?

- No longer in use
  - Missing netmask

- Interesting for some technical “errors”
  - Rich metrics
  - Support for heterogeneous networks (high range for link costs)
  - Multipath

- Are those “errors”, or “features” required to differentiate IGRP from its competitors?

# Metrics (1)

- Based on 4 parameters

| Metric | Allowed values | Example |
|---|---|---|
| B - bandwidth | 1 - $2^{24}$ | 1 = 1.2 kbit/s |
| | In fact, it represents the number of 10ms time slots required to transmit 10K bits | |
| D - delay | 1 - $2^{24}$ | 1 = 10 ms |
| R - reliability | 1 - 255 | 255 = 100% |
| L - load | 1 - 255 | 255 = 100% |

- But… how do we set those values?

## Metrics (2)

- Bandwidth and Delay: automatically associated to each link
  - Basically, B is inversely proportional to the link bandwidth, while D is directly proportional to it
    - In fact, D considers only the “transmission delay”, ignoring other components (e.g., propagation delay, queuing delay)
  - The network admin can change those values manually
    - Not a good idea to do so, though
- Example
  - Ethernet  B=10000, D=100
  - Direct link @64 kbit/s  B=64, D=2000
- R and L are difficult to determine, and they can change with rather high frequency

# Metrics (3)

- In addition, 5 coefficients (k1-k5) allow to tune the importance of each parameter
  - E.g., privilege bandwidth against the delay
- The cost of each link is calculated as follows:
  - When k5 = 0 (default):
    - Cost= $(10^7 / B)$ [k1+ k2 / (256-L)] + k3 D
  - When k5 <> 0 :
    - Cost = Cost x [ k5 / (R + k4) ]
- Default values: k1=k3=1; k2=k4=k5=0
- IGRP commands require the specification of a TOS, but in practice TOS-based routing was never implemented in that protocol
  - TOS must be zero

# Problems with IGRP rich metrics (1)

- While B,D,R,L look good, they originate several problems
- 1) Difficult to understand the best paths on the network
  - Humans look at the network topology and measure the distance in “hop counts”
  - Not easy to determine which is the best path when a more sophisticated metric is adopted
- 2) Difficult to understand how to tune k1-k5
  - What does it happen to my network when parameters are changed? Which values do we have to give them in order to obtain the wanted behavior?

# Problems with IGRP rich metrics (2)

- 3) Possible the presence of high number of transients
  - Some metrics (e.g., load) are not very stable, hence forcing the network to continuously adapt their paths to those changing costs
    - Changing the state of the network → necessity to update the routes → transients → black holes, bouncing effects, etc.
    - Other side effects of continuous routing updates
      - More routing traffic
      - More CPU resources dedicated to routing protocols
    - A stable network (although not 100% optimized) is preferred
- Lesson learned
  - We have to be careful to choose what is really important from the technical point of view, without relying on marketing propaganda

# Problems with rich metrics and infinity (1)

- IGRP has a high number of values of cost for some metrics ($2^{24}$)
- Difficult to define the right value of the "infinity"
  - If a link has cost=1000, inf=16000 may look ok
  - What about if I have also very fast links, with cost=1?
    - 16K iterations before reaching infinity
- Other DV protocols (e.g., RIPX, Decnet Phase 3) use two metrics for the cost
  - Hop count (used for the infinity)
  - The other metric (e.g., bandwidth) to compute the paths
  - When the hop count reaches infinity, the path will be unavailable no matter the value of the cost

# Problems with rich metrics and infinity (2)

- IGRP defines infinity is 16777216 ($2^{24}$)
- Slow convergence to infinity when links with small cost are involved
  - The cost of the path increases slowly each round
  - A link that costs “1” takes so much time to reach the infinity threshold in IGRP

# Multipath routing

- Allows multiple routes for the same destination
  - Even if those routes have different costs
  - RIP allows a single route per destination
- Load is distributed proportionally to the cost of the route
  - Two routes at the same cost get 50/50 of the traffic
  - Two routes at cost 2 and 4 respectively get 2/3 and 1/3 of the traffic
- A threshold V is used to discriminate whether we can use two routes or not
  - We can use a secondary route if
    - Cost secondary route <= (V * cost primary route)
  - V can be modified by the network manager

## Problems with multipath routing

- Traffic may enter in a loop when different paths are chosen by two routers
  - One can choose the primary path (main route) and the other the secondary path (second route)
- Coefficient “V” is set to 1 in the latest version of IGRP in order to prevent those issues
  - Only equal cost multipath routing allowed

# Stability techniques

- Very similar to RIP
- Triggered updates
  - Sent each time the cost changes for more than 10%
  - Avoids frequent reconfiguration of the network
- Split Horizon (without poisoned reverse)
- Path Hold Downs
- Route poisoning
  - Used in more recent implementation of the IGRP instead of the Path Hold Downs

# Timers

- Very similar to RIP, but with different default values
  - update timer 90s
  - invalid timer 3 x update
  - flush timer 7 x update
  - hold down timer (3 x update) + 10s

# Enhanced IGRP

- Another proprietary Cisco routing protocol
- Loosely based on its ancestor IGRP
  - Kept the “IGRP” name mainly for marketing reasons
  - Configuration is almost the same
    - Hence, from the perspective of the network admin, it looks definitely similar to IGRP
- IGRP had to be abandoned because of the lack of the network mask

# EIGRP in a nutshell (1)

- Distance Vector
  - IGRP with the usual metrics
  - Infinity is now $2^{32}$ (4294967296)
- \+ Diffusing Update Algorithm
  - Loops free (but black holes still possible)
  - Faster convergence (no count-to-infinity)
- \+ Reliable Updates
  - Decreased routing traffic
  - “Hello” protocol needed

# EIGRP in a nutshell (2)

- + Miscellaneous
  - Network masks (IGRP was classful)
  - Multiprotocol
  - Less processing-intensive
    - Hello messages are processed very easily, while DV required more extensive computation

- More details later

# Most important changes in EIGRP (1)

- Adoption of the DUAL algorithm (in addition to Distance Vector) in order to prevent routing loops
  - Main idea: alternate paths are chosen only when we are sure they cannot bring loops into the network
    - Otherwise, better to put the route in hold down
  - This algorithm is the key point to guarantee a loop-free transient

# Most important changes in EIGRP (2)

- Decouples two functions that are needed in a routing algorithm
  - Neighbor discovery
    - New "hello" protocol
  - Route update
    - Propagation of the distance vector

# Most important changes in EIGRP (3)

- No periodic updates of the DV
  - Possible because of the decoupling between neighbor discovery and routing updates
  - The new DV is sent only when something changed in the network
    - A router keeps the DV of all its neighbors, as usual
  - DV must be sent through a reliable protocol
    - EIGRP implements a simple mechanism based on an ACK that is sent back from the neighbor when it receives a new DV
  - Advantages
    - Reduced routing traffic (hello packets much smaller than DVs)
    - Reduced CPU load (hello packets are much simpler to process than DV)
  - Consequently, EIGRP can increase the frequency of hello messages
    - Faster fault detection, hence faster convergence

# Packet types

- Hello - multicast, for the “hello” protocol
- Update – distance vector, sent by routers in “passive” state
  - unicast when sent to a specific neighbor (e.g., a new router)
  - multicast when the update has to be propagated to all neighbors (e.g., in case of a topology change)
- Acknowledgement - unicast, to confirm the receipt of an update
- Query - multicast/unicast: sent by a router in “active” state, and corresponds to the beginning of the diffusion process
- Reply - multicast/unicast: reply to a query
- Request – send by a router to get the DV from its neighbors (e.g., used by a router when it starts up)

# Neighbor Table

- Data structure in each EIGRP router
- Each entry contains:
  - Address of the neighbor (learned in the hello message)
  - Hold time (learned in the hello message)
  - Network interface
  - Round-trip timer
  - Additional info for the Reliable Transfer Protocol

# Topology Table

- Contains all the destinations announced by the neighbors of the current router
- Each entry contains:
  - Destination address
  - List of
    - Neighbors that address the destination
    - Associated cost

# State of the routes

- Routes in EIGRP routing table can be either in
  - Passive state: the route is actually used
  - Active state: the route is currently being recalculated
    - The router is executing the DUAL process and is blocking that route
- A route in Active state cannot be updated till it moves into Passive state

# Route Tagging

- EIGRP can distinguish between internal and external routes
  - Internal: the route has been originated by the EIGRP process (either on the local router, or on another connected to the same domain)
  - External: the route has been generated by another routing protocol (or it is a static route, redistributed into EIGRP)
- External routes are tagged with
  - Router ID
  - Destination Autonomous System
  - ID of the originating protocol
  - Metric used in the redistribution process
  - Tag, defined by the network admin
  - A one-bit flag for the default route

# Conclusions

- EIGRP is widely used
  - Despite being proprietary
  - Despite being distance vector
  - Despite not being hierarchical
- Very good protocol, thanks to the many improvements compared to IGRP
  - Not just an IGRP with netmask