Routing Information Protocol

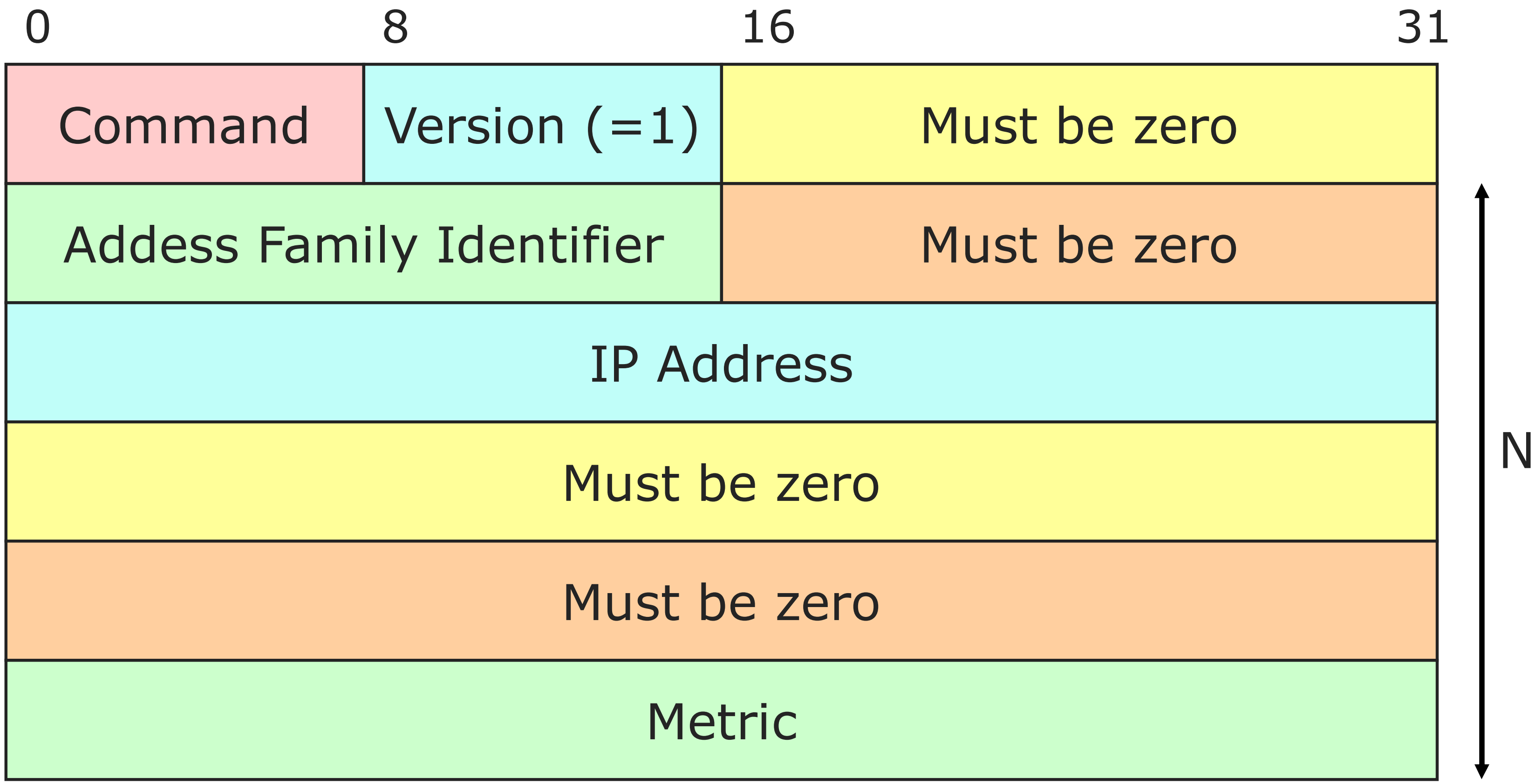
Fulvio Risso

Politecnico di Torino

RIP (Routing Information Protocol)

- First routing protocol on the Internet
  - Used by the “routed” UNIX routing daemon
- Defined in RFC 1058 in 1988
- Based on Distance Vector
- Very simple metric (hop count)
  - Suitable for small, stable and homogeneous networks
- Many implementation available
  - Often different from the official specs
    - No split horizon, other with SH with poisoned reverse, others with route hold down, etc
  - Interesting how the protocol converges even in presence of different implementations
    - May increase the convergence time, but it will converge anyway

Packet format (1)



Packet format (2)

- Command
  - Response: normally used to transmit the distance vector
  - Request: used only by a new router after starting up, to notify its neighbors of its presence
    - Speeds up the convergence time, as neighbors send their DV back without having to wait for the timeout
- AFI
  - Set to 2 in IP
  - RIP was never used in other protocol families
- The last 5 fields can be repeated up to 25 times
  - Max Packet Length: 512 bytes (25*20+4 RIP, 8 UDP)

Packet format (3)

- Encapsulation: RIP --> UDP --> IP
  - Port 520
  - Considered as a security mechanism (ports < 1024 can be used only under administrative privileges)
    - Obviously this is no longer valid now
  - RIP is a layer-3 protocol, despite the encapsulation it uses
    - Much simpler to create the software using TCP/UDP sockets than programming at lower layers
- Sent to broadcast IP address (255.255.255.255)
  - Every devices (also hosts) will receive it
  - Used in the past to let the hosts know the routes
  - No longer a best practice
    - In fact, usually routing protocols are disabled on the host side, to protect them from malicious attacks

Timers (1)

- Routing update timer (default 30 s)
  - Interval between two updates
  - In fact, DV can be sent with a delay between 25 and 35 seconds to avoid router synchronization
    - A fixed interval of 30 seconds was discovered to be problem in the early Internet
- Route invalid timer (default 180 s)
  - Time required for a route to become unreachable if not refreshed
  - Especially useful to detect a missing connectivity toward a neighbor when the “link down” signal is not available

Timers (2)

- Route flush timer (default 240 s)
  - Time required for a route to be cancelled from the routing table
    - A route is not deleted at the expiration of the route invalid timer
    - We need to wait for 60 additional seconds, in which the route is kept at cost equal to infinity (and sent in DVs)
    - Needed to be sure that all out neighbors receive the notification of the missing route
- Hold down timer (default 180s)
  - Only on Cisco devices

Infinity

- RIP defines the Hop Count Limit equal to 16
  - Destinations whose distance is larger than 15 are considered unreachable
  - This does not mean that the network cannot have more than 15 routers, of course
    - Although, the longest path cannot exceeds this threshold
- Limits the well-known "count to infinity" problem of Distance Vectors algorithms

Triggered updates

- RIP uses triggered updates
  - A reduced DV is sent "immediately" whenever a change in the network is detected
    - Only changed routes are included in that DV, not the entire routing table
  - Helps in speeding up the convergence time
- Does not reset the update timer
  - Standard DVs are sent with the well-known interval
- Sent with a delay between 1 and 5 seconds
  - To group several events in the same DV
  - To avoid too frequent updates
  - To avoid synchronization

Split Horizon

- DV sent by a router are different on the various links
  - Each router does not send back the routes it learned from that neighbor
  - “Never tell the same joke to someone who just told it to you”
- Usually coupled with “poisoned reverse”
  - Reverse routes are sent with cost equal to infinity
  - More robust in real implementations
- Useful on some simple loops, but it does not prevent more complex loops (between three nodes)

Route poisoning

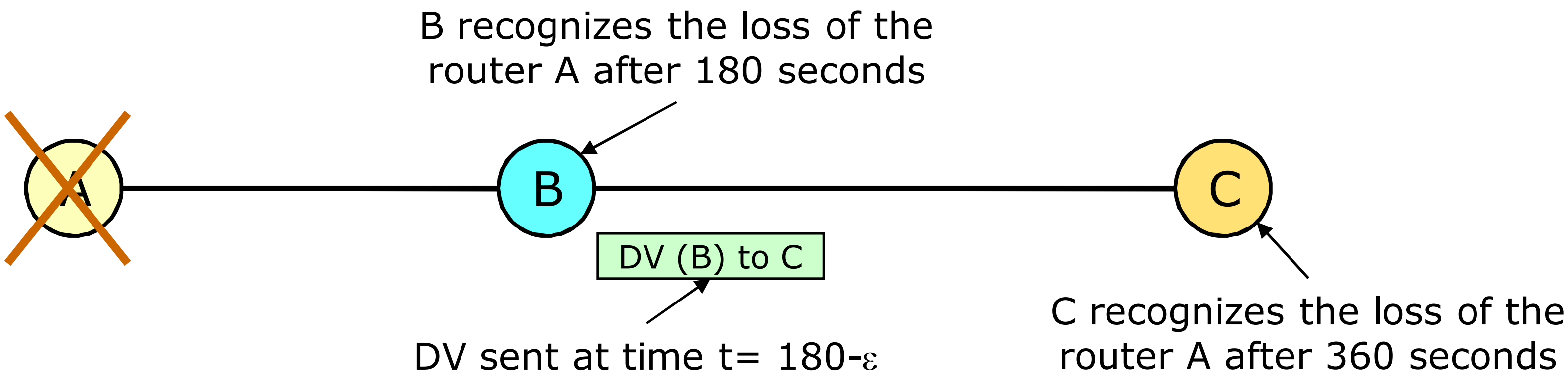
- The same idea of the “Split Horizon with Poisoned Reverse”
  - “Bad news is better than no news”
- Routes that are unreachable are propagated at cost = infinity
  - A network directly connected to the current router is now unreachable
  - A destination blocked by the Path Hold Down algorithm may be propagated at cost infinity

Route Hold Down

- Not in RFC 1058, but present in the Cisco implementation
- If the cost of a route increases (becomes infinity?), that route is no longer modified till the expiration of the HoldDown timer
  - “Let the rumors calm down and wait for the truth”
  - The router still uses the “old” route
    - We may have a black hole, but not a routing loop
- After the HoldDown, the router will accept any modification to that route

An important mistake: the missing “age”

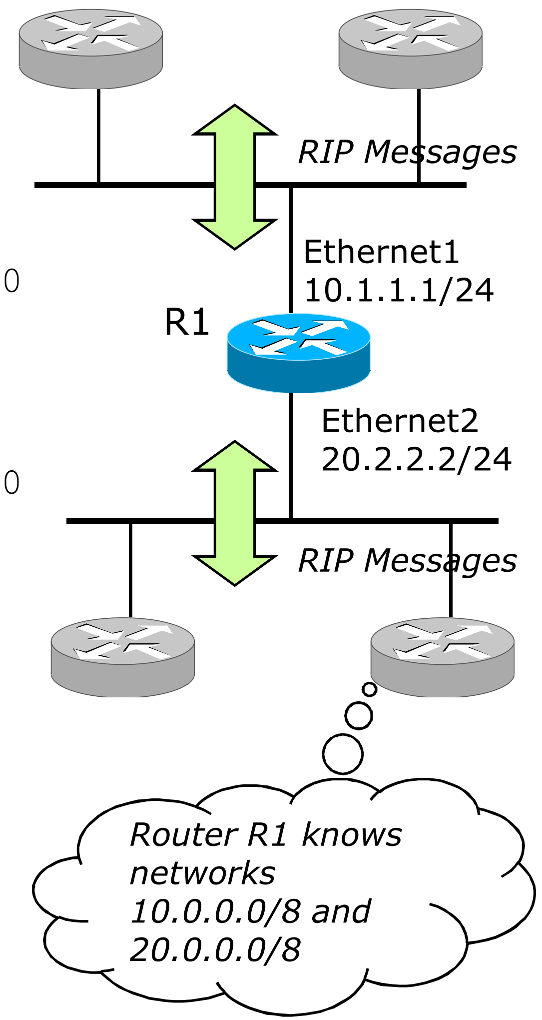
- RIP does not associate an “age” to the route in the distance vector
- Example
  - Let’s suppose no triggered updates, no route poisoning
  - Time=0: fault on router A
  - Time=180-ε: B send the last DV containing the destination A to C
    - This DV will stay active for 180 seconds in C’s memory
  - Time=2x180: C recognizes that A is no longer reachable



RIP v.1 configuration example

```
R1> enable
R1# configure terminal
R1(config)# interface ethernet1
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# exit
R1(config)# interface ethernet2
R1(config-if)# ip address 20.2.2.2 255.255.255.0
R1(config-if)# exit
R1(config)# router rip
R1(config)# network 10.1.1.1
R1(config)# network 20.2.2.2
R1(config)# end
R1#
```

Better to use network address instead



RIP version 2

Overview

- RFC 2453
- Compatible with RIP v.1
- Uses the many "unused" fields in RIP v.1 messages
- Associates the network mask to the network destination
  - This was the main problem of RIP v.1, which was defined when the classful addressing was used
- Additional improvements (not very significant, though)
  - Handles external routes ("route tag")
  - Authentication (very weak)
  - Routing optimization ("next hop")
  - Traffic optimization (multicast)
  - RFC 1388 (original RIPv2 spec) proposed to handle more routing domains on the same router, but this was later removed in RFC 1721 (and Cisco does not allow to configure that feature)

Packet format (RFC 2453)

0	8	16	31
Command	Version (=2)	Must be zero	
Addess Family Identifier		Route Tag (RIP1: MBZ)	
IP Address			
Subnet Mask (RIP1: MBZ)			
Next Hop (RIP1: MBZ)			
Metric			

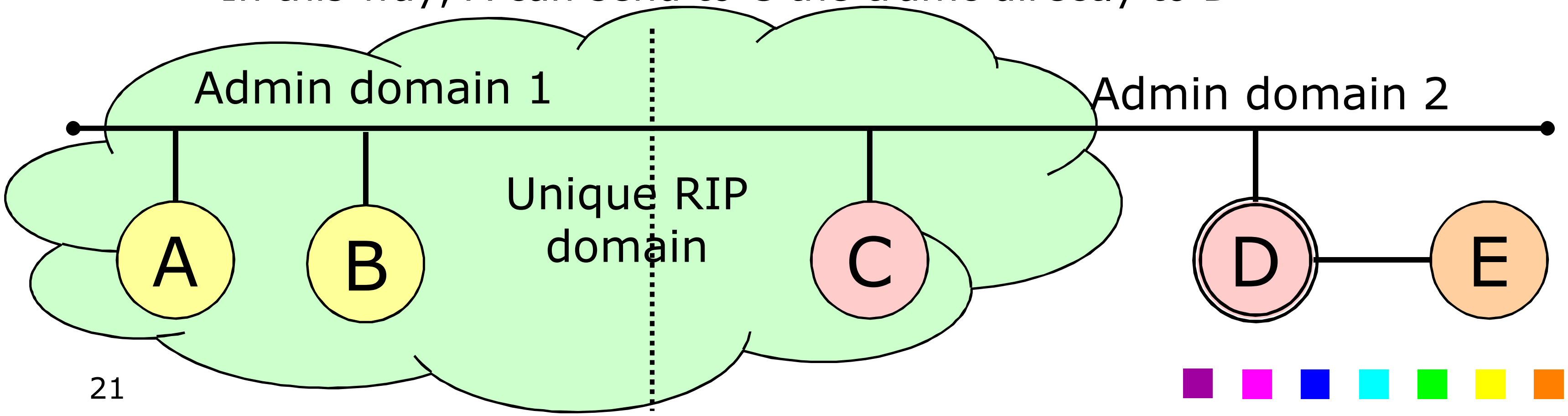
N (rows Addess Family Identifier through Metric)

Route Tag

- Used to mark routes that are external to the current RIP domain
  - The route tag value is transparent (i.e., forwarded unmodified) to RIP
- Potentially used when the RIP domain is connected to the external world with BGP
  - E.g., the route tag can keep the Autonomous System number where this route was learned

Next Hop

- Optimizes the routing when multiple RIP routers are on the same LAN, and (for some reason) the best next hop toward a destination is not know by all the routers
- Example: router A has to send data to destination E, which does not belong to the same RIP domain
  - Router A does not know router D
  - Router A should send data traffic to C, which is then forwarded to D
  - Solution: B can advertise D as next hop for destination E
  - In this way, A can send to C the traffic directly to D



Authentication

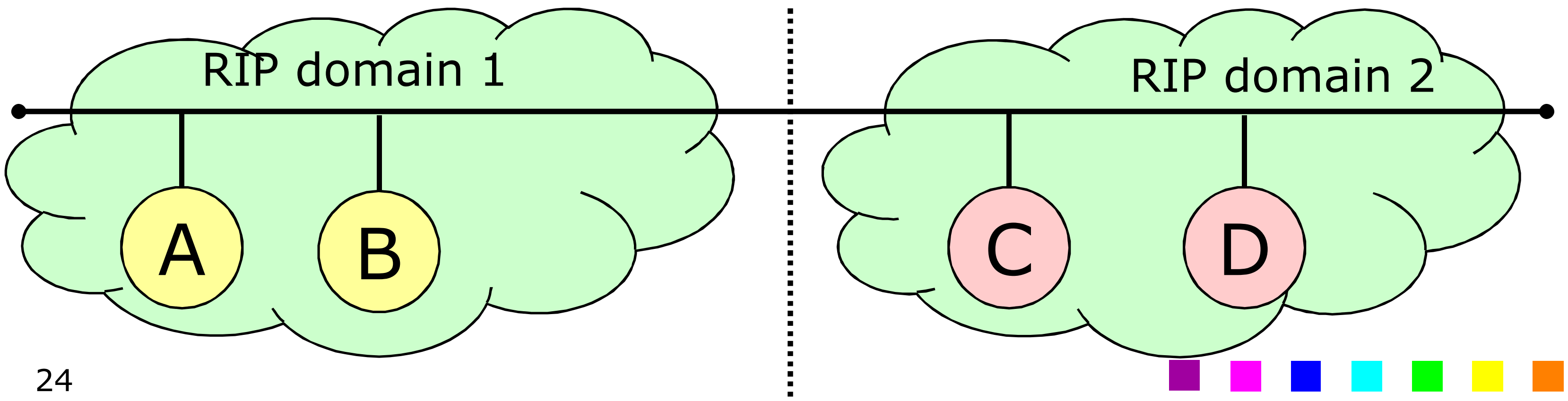
- Simple authentication, password based
  - More complex authentications (e.g., MD-5) are not possible because of the lack of space in the RIP message
    - Message cannot be extended in order to keep compatibility with RIP v.1
- Authentication message
  - AFI set to 0xFFFF
  - Authentication type (only "Simple password" has been defined)
  - 16 bytes of authentication follows
- RIP v.1 routers do not recognize this AFI and skip the message

Multicast

- RIP v.1 sends DV in broadcast
  - All the entities (including hosts) have to process those messages
- RIP v.2 defines an IP multicast address for that
  - 224.0.0.9
  - Possible to configure the protocol to use the broadcast address in order to maximize the compatibility with RIP v.1

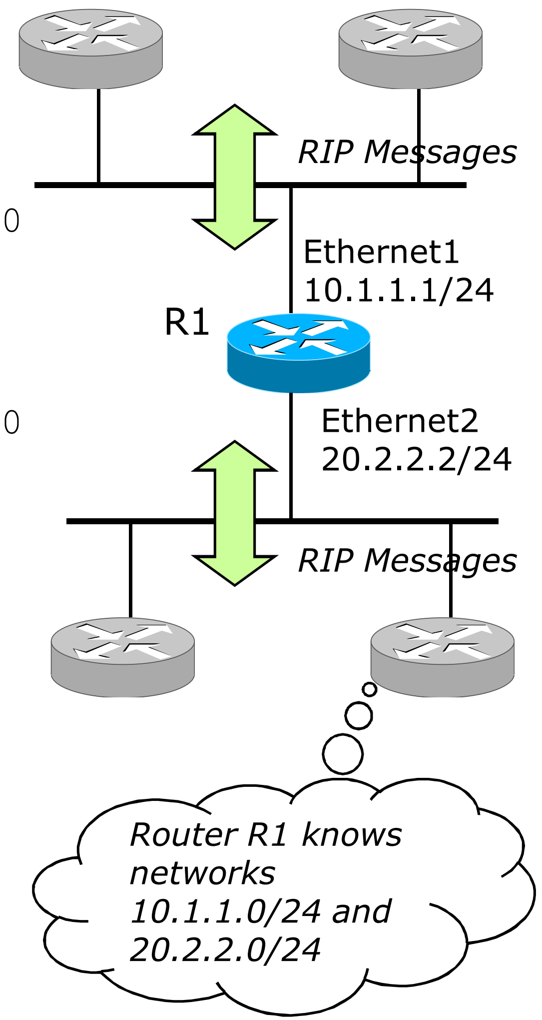
No support for multiple routing domains

- RIPv2 does not support different routing domains **on the same LAN**
- A RIP message generated by router A will be received by routers in domain 2 as well
- This topology can be achieved by a special (manual) configuration
  - ACL filtering in order to discard unwanted RIP messages coming from specif sources
  - Forcing RIP to send unicast messages instead of multicast



RIP v.2 configuration example

```
R1> enable
R1# configure terminal
R1(config)# interface ethernet1
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# exit
R1(config)# interface ethernet2
R1(config-if)# ip address 20.2.2.2 255.255.255.0
R1(config-if)# exit
R1(config)# router rip
R1(config)# version 2
R1(config)# no auto-summary
R1(config)# network 10.1.1.1
R1(config)# network 20.2.2.2
R1(config)# end
R1#
```



RIPng

Overview

- Same operating principles as RIPv2
  - Distance vector, radius of 15 hops, split horizon and poison reverse
  - Based on RIPv2
- Updated features for IPv6
  - IPv6 prefix, next-hop IPv6 address
  - Uses the ALL-RIP-ROUTERS multicast group as the destination address for RIP updates
    - Multicast group FF02::9
  - Uses IPv6 for transport

RIPng in Cisco

- Main improvements in the Cisco command line interface

- Support for multiple RIP instances
  - Commands support the <TAG> info
  - Please note that support for multiple instances was present in RIPv2, but not configurable on Cisco routers

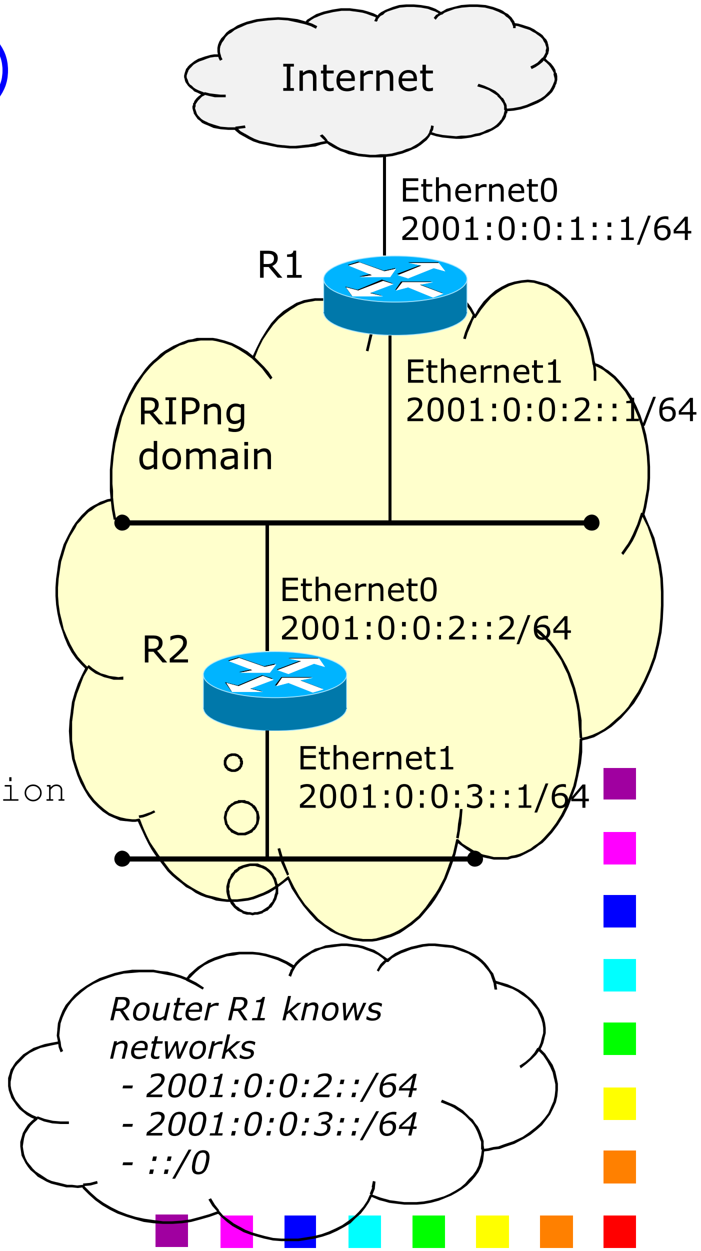
- Per-interface configuration
  - The “network” command is no longer used

Cisco RIPng Reference

- Global configuration command:
  - **ipv6 router rip <tag>**: Starts a RIP process. Enters the router sub-command context
- Router subcommands:
  - **redistribute static|bgp|rip <tag>**: Redistribute route entries from other routing processes
- Interface subcommands:
  - **ipv6 rip <tag> enable**: Configure RIP on that interface
  - **ipv6 rip <tag> default-information originate**: Originate the default route (::/0) and send it in updates
- Exec commands:
  - **show ipv6 rip**: Display status of the various RIP processes
  - **debug ipv6 rip**: Debug RIP. Display RIP packets sent and received

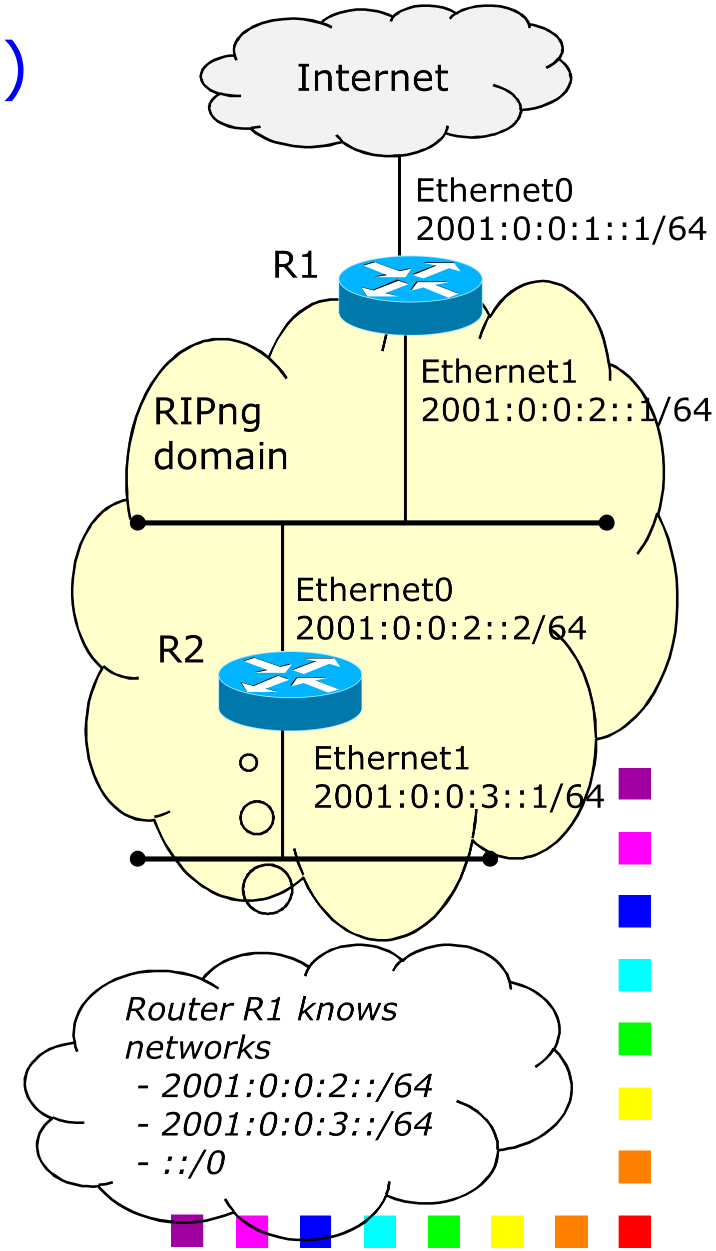
RIPng configuration example (1)

```
R1> enable
R1# configure terminal
R1(config)# ipv6 router rip Domain0
R1(config)# interface ethernet0
R1(config-if)# ipv6 address 2001:0:0:1::1/64
R1(config-if)# exit
R1(config)# interface ethernet1
R1(config-if)# ipv6 address 2001:0:0:2::1/64
R1(config-if)# ipv6 rip Domain0 enable
R1(config-if)# ipv6 rip Domain0 default-information
                  originate
R1(config-if)# exit
R1(config)# ipv6 route ::/0 2001:0:0:1::2
R1(config)# end
R1#
```



RIPng configuration example (2)

```
R2> enable
R2# configure terminal
R2(config)# ipv6 router rip Domain0
R2(config)# interface ethernet0
R2(config-if)# ipv6 address 2001:0:0:1::1/64
R2(config-if)# ipv6 rip Domain0 enable
R2(config-if)# exit
R2(config)# interface ethernet1
R2(config-if)# ipv6 address 2001:0:0:2::1/64
R2(config-if)# ipv6 rip Domain0 enable
R2(config-if)# exit
R2(config)# end
R2#
```



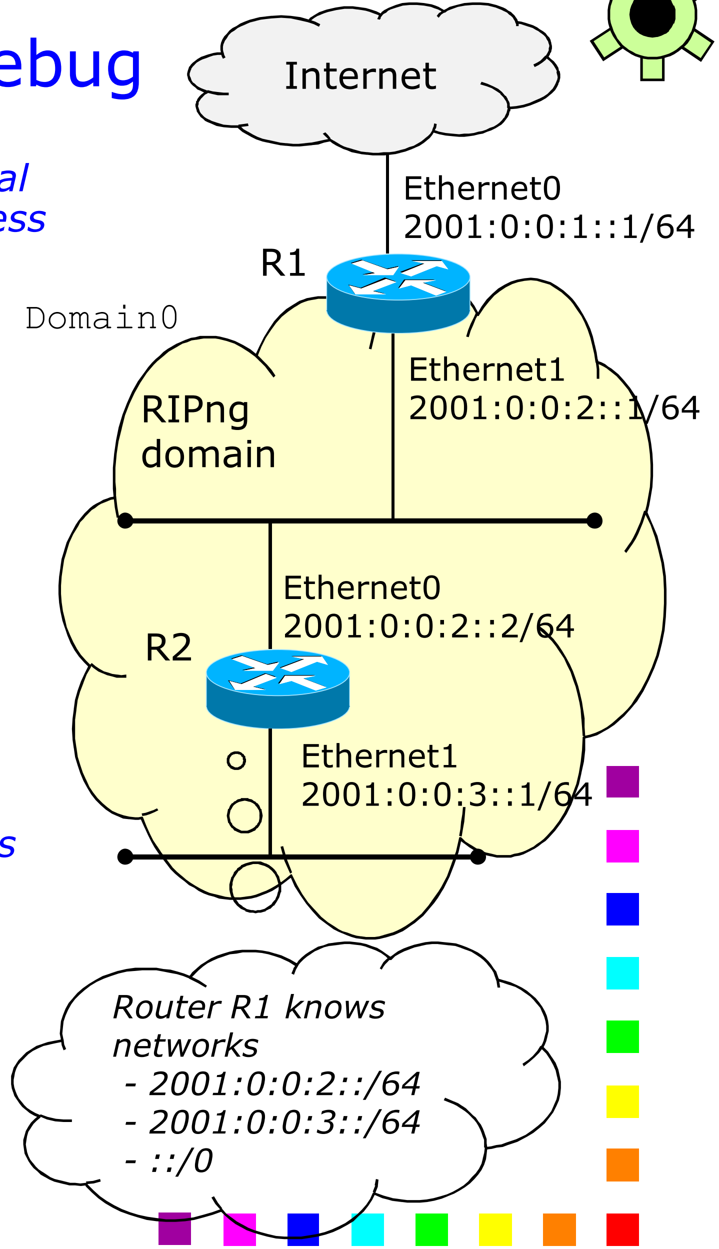
RIPng configuration example: debug

```
R2> enable
R2# debug ipv6 rip
RIPng: Sending multicast update on Ethernet0 for Domain0
       src=FE80::260:3eff:fe47:1530
       dst=FF02::9 (Ethernet0)
       sport=521, dport=521, length=32
       command=2, version=1, mbz=0, #rte=1
       tag=0, metric=1, prefix=::/0
R2#
```

Link-local src address

Multicast all rip-routers

Default route



Conclusions

- RIP is a very simple protocol, slow convergence
- Good choice anyway for networks
  - Small networks (a few routers)
  - Stable networks (a few transients)
    - Convergence problems are noticeable only during transients
  - Homogeneous links
    - Cannot differentiate costs on different links (e.g. link speed)
- Advantages
  - Simple (a few CPU cycles/memory required; simple implementation)
  - Available on a wide range of devices (also the ones that you can buy for a few bucks in electronic shops)
  - Very simple to configure (e.g., no subdomains such in OSPF, no need to know anything about Distance Vector)