Chapter-5: IP Routing & Distance Vector Family

Deep Medhi and Karthik Ramasamy

September 2007

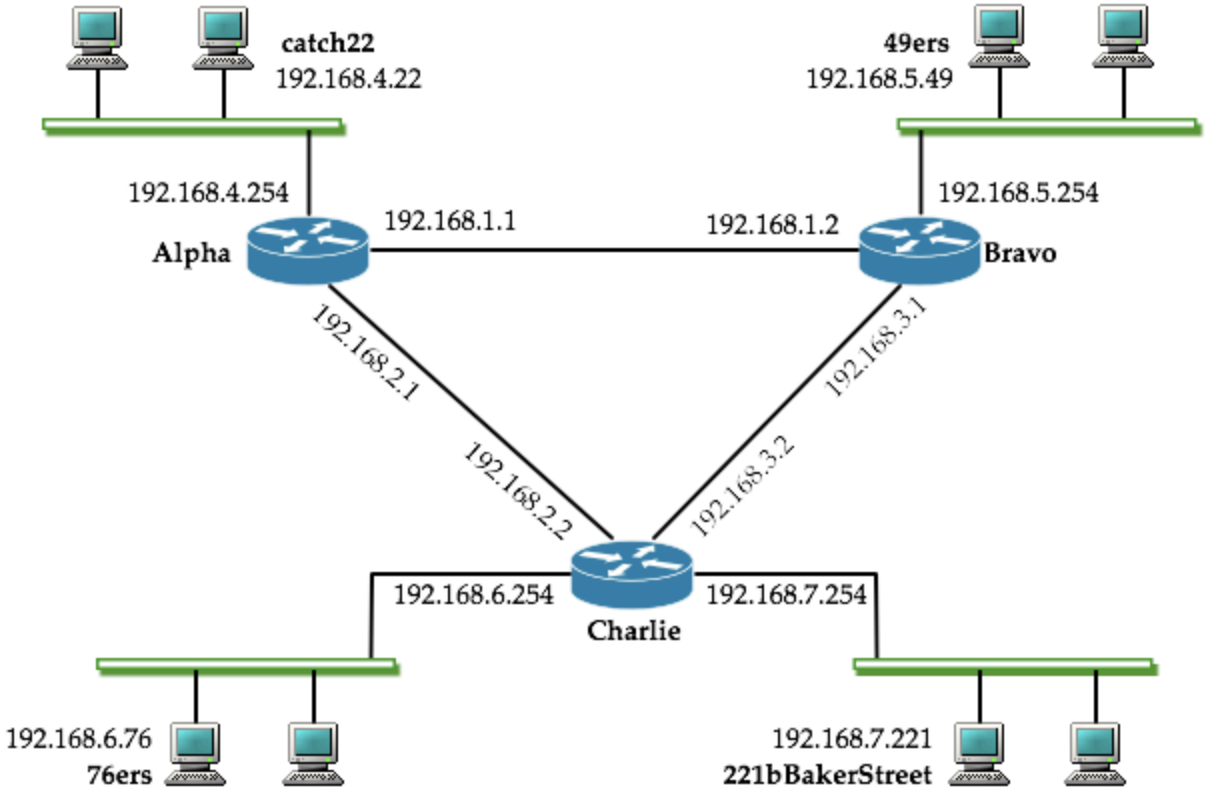
http://www.NetworkRouting.net

Routing in IP Networks

- Address plays a critical role; routing based on destination address with netmask (subnet)
  - See Chapter-1 about netmasking
- How does the routing table look like at a router?
  - What type of information it maintains

An example Network

- Several subnets off routers; assume mask to be /24
- Each router interface defined with an IP address (also on /24 mask)



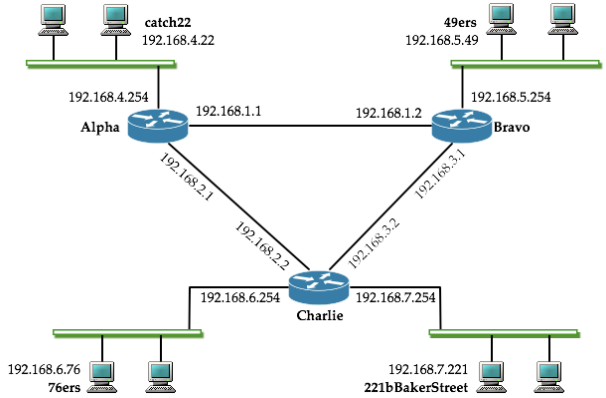


TABLE 5.1 Routing table at each router for the network shown in Figure 5.1.

Router: Alpha		Router: Bravo		Router: Charlie	
Network/Mask	Next Hop	Network/Mask	Next Hop	Network/Mask	Next Hop
192.168.1.0/24	direct	192.168.1.0/24	direct	192.168.1.0/24	192.168.2.1
192.168.2.0/24	direct	192.168.2.0/24	192.168.1.1	192.168.2.0/24	direct
192.168.3.0/24	192.168.1.2	192.168.3.0/24	direct	192.168.3.0/24	direct
192.168.4.0/24	direct	192.168.4.0/24	192.168.1.1	192.168.4.0/24	192.168.2.1
192.168.5.0/24	192.168.1.2	192.168.5.0/24	direct	192.168.5.0/24	192.168.3.1
192.168.6.0/24	192.168.2.2	192.168.6.0/24	192.168.3.2	192.168.6.0/24	direct
192.168.7.0/24	192.168.2.2	192.168.7.0/24	192.168.3.2	192.168.7.0/24	direct

- Routing Table stores by destination address blocks (not based on destination router address!)

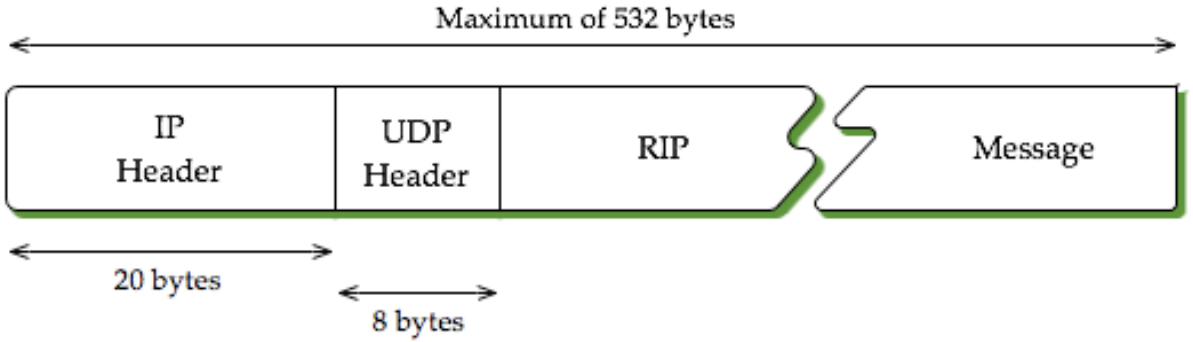
How is routing table formed?

- To build such a routing table, the IP network uses a routing protocol mechanism to communicate information about reachable destination IP address blocks
  - Different protocols available for intra-network and inter-network
  - These protocols use the same IP infrastructure Thus, routing protocols need to rely on the TCP/IP protocol stack for making this possible; e.g.,
    - RIP uses UDP to pass to routing protocol application
    - OSPF passes directly to protocol application
    - BGP uses TCP to pass to routing protocol application
- Information received from neighbors is used by the routing computation module in order to create the routing tabel

RIPv1

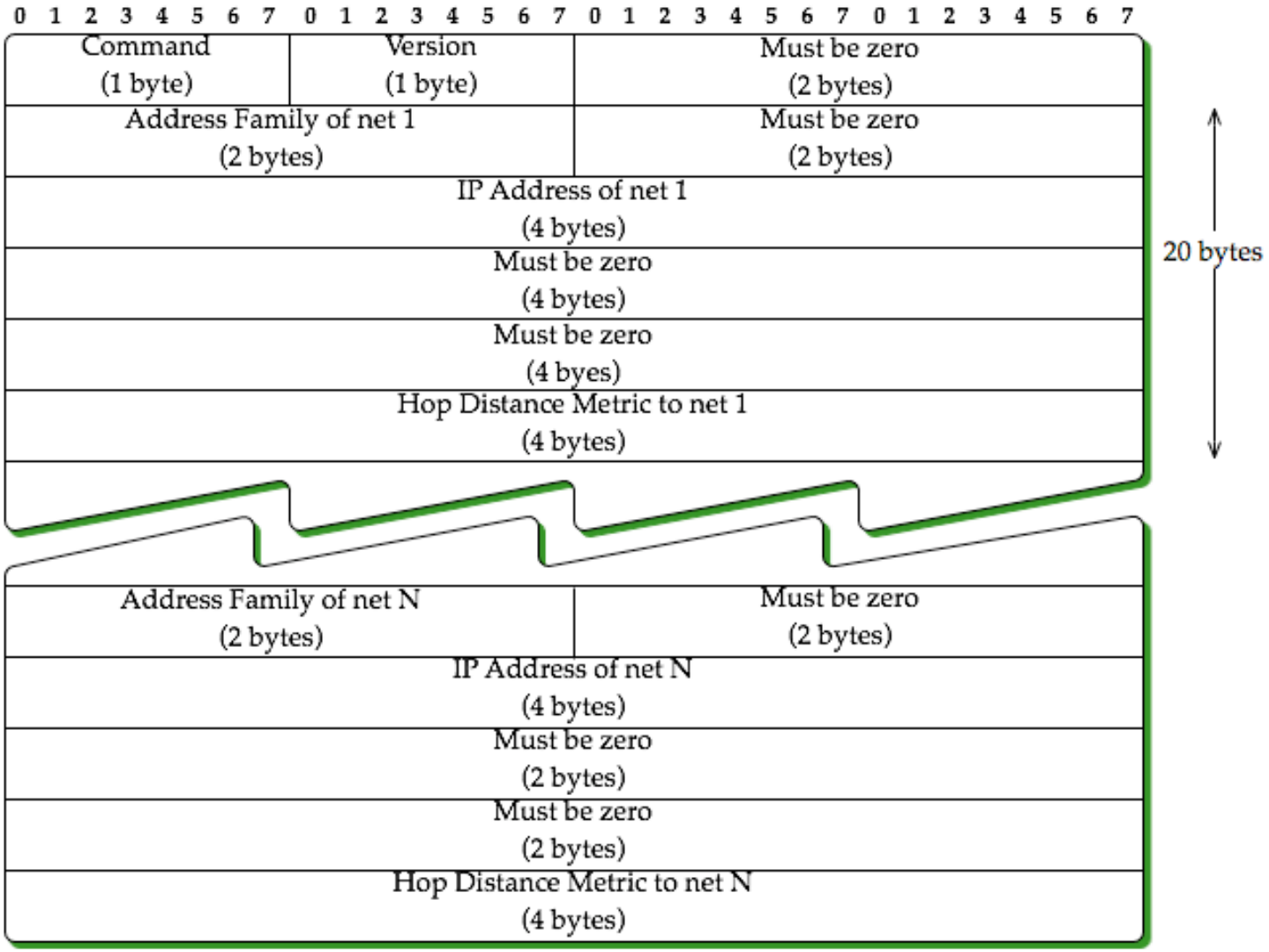
- Routing Information Protocol (RIP), v1
  - Is based on the Distance Vector protocol concept
  - Specified in RFC 1058 (in 1988)
    - However it was around since early 1980s
  - Use UDP as the transport protocol
    - Port # 520 has been the default port for RIP communication
  - Has evolved to RIPv2 [see RFC 2453].
  - RIPng for IPv6

- RIP messages are carried in UDP packets



- A RIPv1 message has a common 4 byte header, followed by 20-byte message for each destination address blocks (commonly referred to as ‘route’)
- An UDP packet has 512 byte limitation; thus, it can hold up to a max of 25 routes.

RIPv1 packet format: details



RIPv1 protocol fields

- *Version* (1 byte): This field indicates the RIP protocol version. This is set to 1 for RIPv1. If this field happens to be zero, the message is to be ignored.
- *Address family identifier* (2 bytes): This field identifies the address family. Set to 2 for the IP address family. There is a special use case when this field is set to zero; see command field later.
- *IP address* (4 bytes): This is the destination network, identified by a subnet or a host.
- *Metric* (4 bytes): This is based on hop count; it is a number between 1 and 16 where 16 means unreachable or infinity.

[cont'd]

- *Command* (1 byte): This field is used for different command sets in a RIPv1message.
- Two used: *request* and *response*;
  - Request: used by a router to request a neighboring router for distance vector information.
    - If the entire routing table is desired, a request message (referred to as "request-full") is sent where the address family identifier is set to 0 and the metric to infinity; the response follows a split horizon (see Section 3.3.3).
    - If responses are sought for a set of address blocks (referred to as "request-partial"), the request flag is set, the address family identifier is set to IP, and the addresses are listed; the responding router sends a response to all addresses listed (no split horizon is done in this case).

RIP: General Operation

(Assumes familiarity with Distance Vector Protocol, Section 3.3)

- General packet handling: if any of the must-be-zero fields have nonzero values anywhere or if the version field is zero, the packet is discarded.
- Initialization: when a router is activated and it determines that all the interfaces are alive, and it broadcasts a request message that goes to all interfaces in the "request-full" mode. The neighboring routers handle responses following the split horizon rule. Once the responses are received, the routing table is updated with new routes the router has learned about.
- Normal routing updates: in the default case, this is done approximately every 30 sec ("*Autoupdate timer*") where updates are broadcasted with command fields set to the response mode; a large variation is added to avoid the *pendulum effect* (see Section 3.3.3, pp. 72-73)
- Normal response received: the routing table is updated by doing the distributed Bellman–Ford step; only a single best route is stored for each destination.

[cont'd]

- Triggered updates:
  - if the metric for an addressable network changes, an update message is generated containing only the affected networks.
- Route expiration:
  - if an addressable network has not been updated for 3 min ("*expiration timer*") in the default case, its metric is set to infinity and it is a candidate for deletion. However, it is kept in the routing table for another 60 sec; this extra time window is referred to as *garbage collection* or *flush* timer.

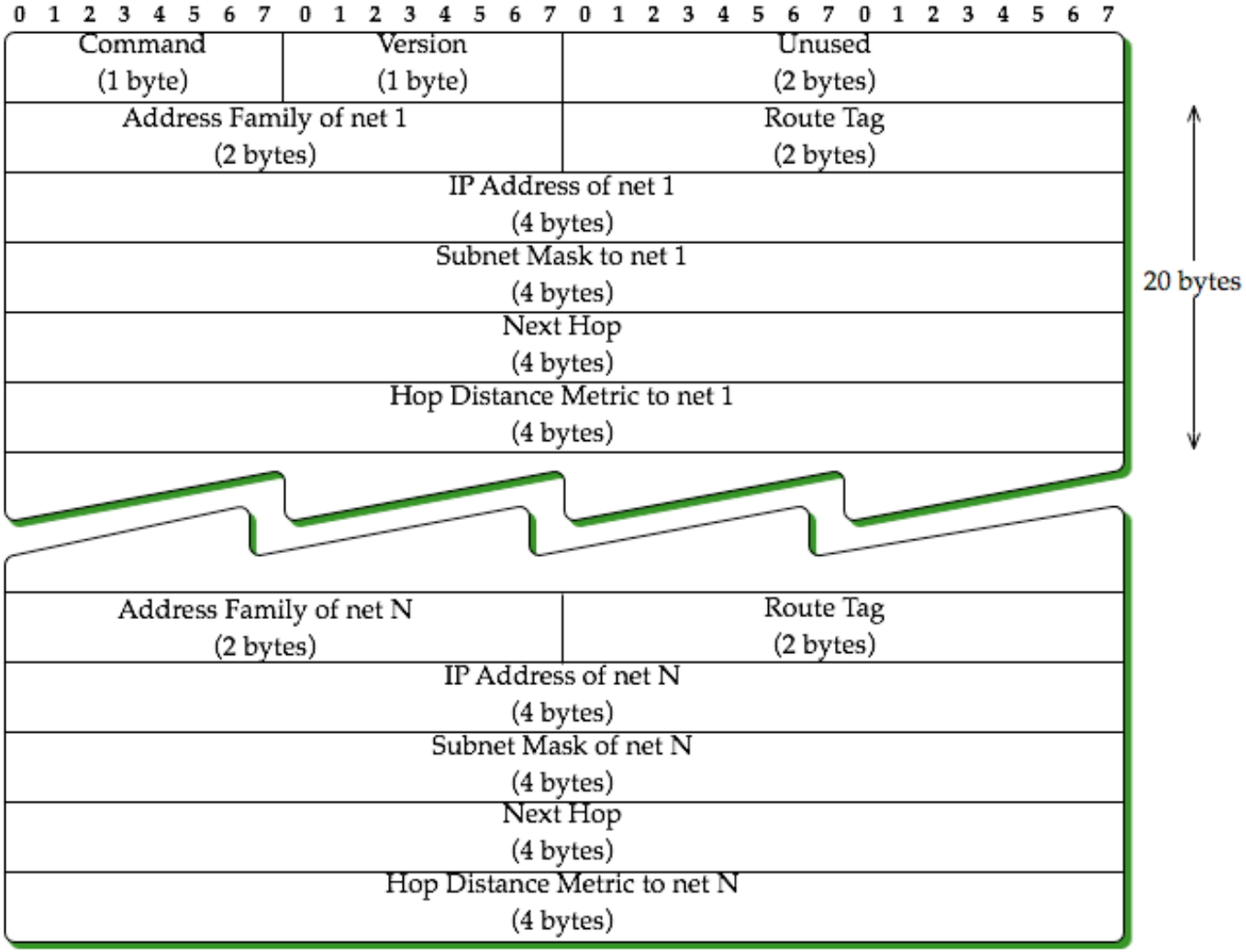
RIPv1

- Can be used only for classful address (Class A, Class B, Class C)
- Useful only for small networks
- Since link cost is based on hop count, can't do traffic engineering
  - Works well if there's plenty of capacity
- Has all the problems of a distance vector protocol (see Section 3.3)

RIPv2

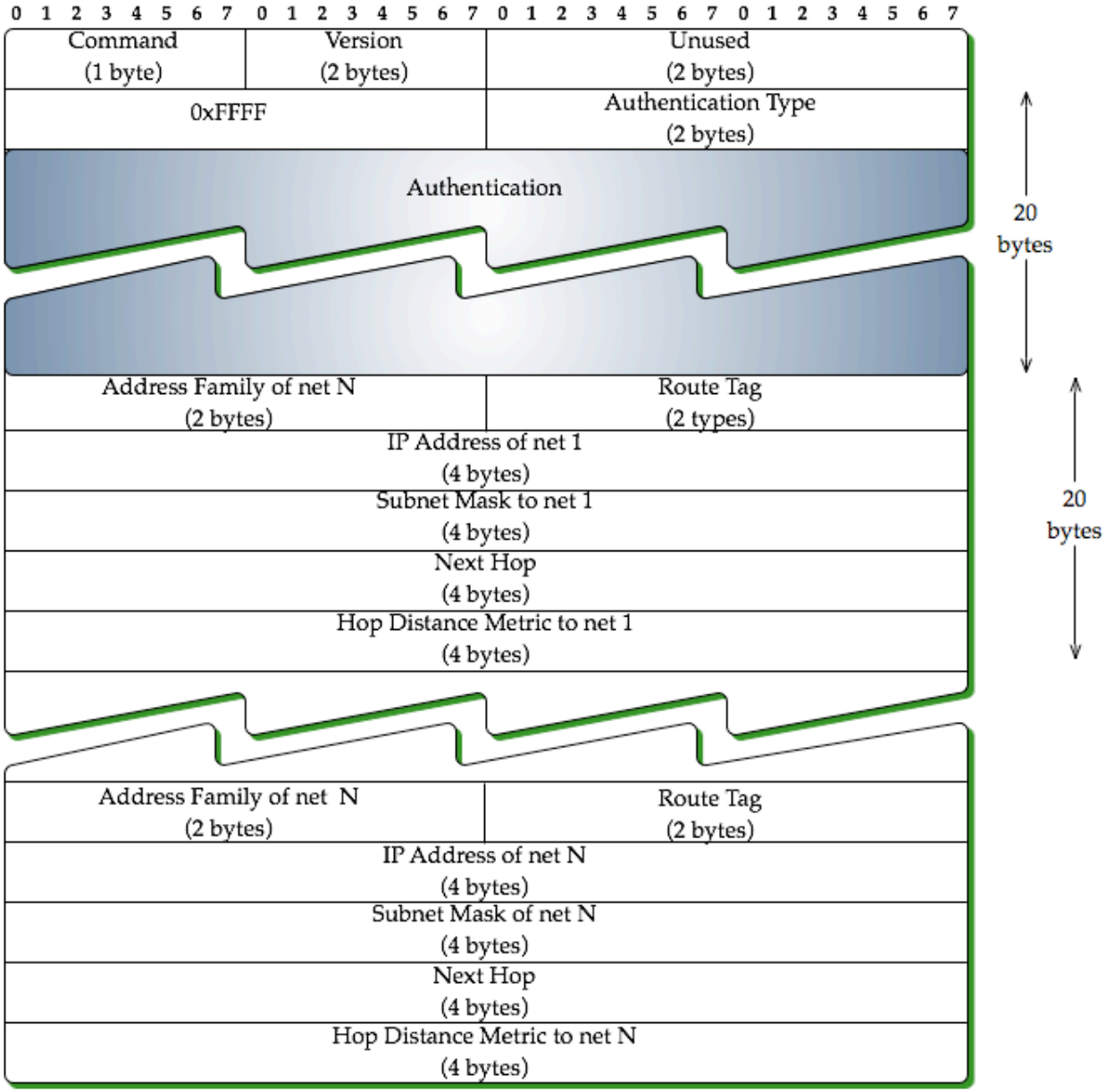
- Enhancement over RIPv1
  - Allows explicit netmasking
  - Authentication is available

RIPv2: packet format



Additional fields in RIPv2

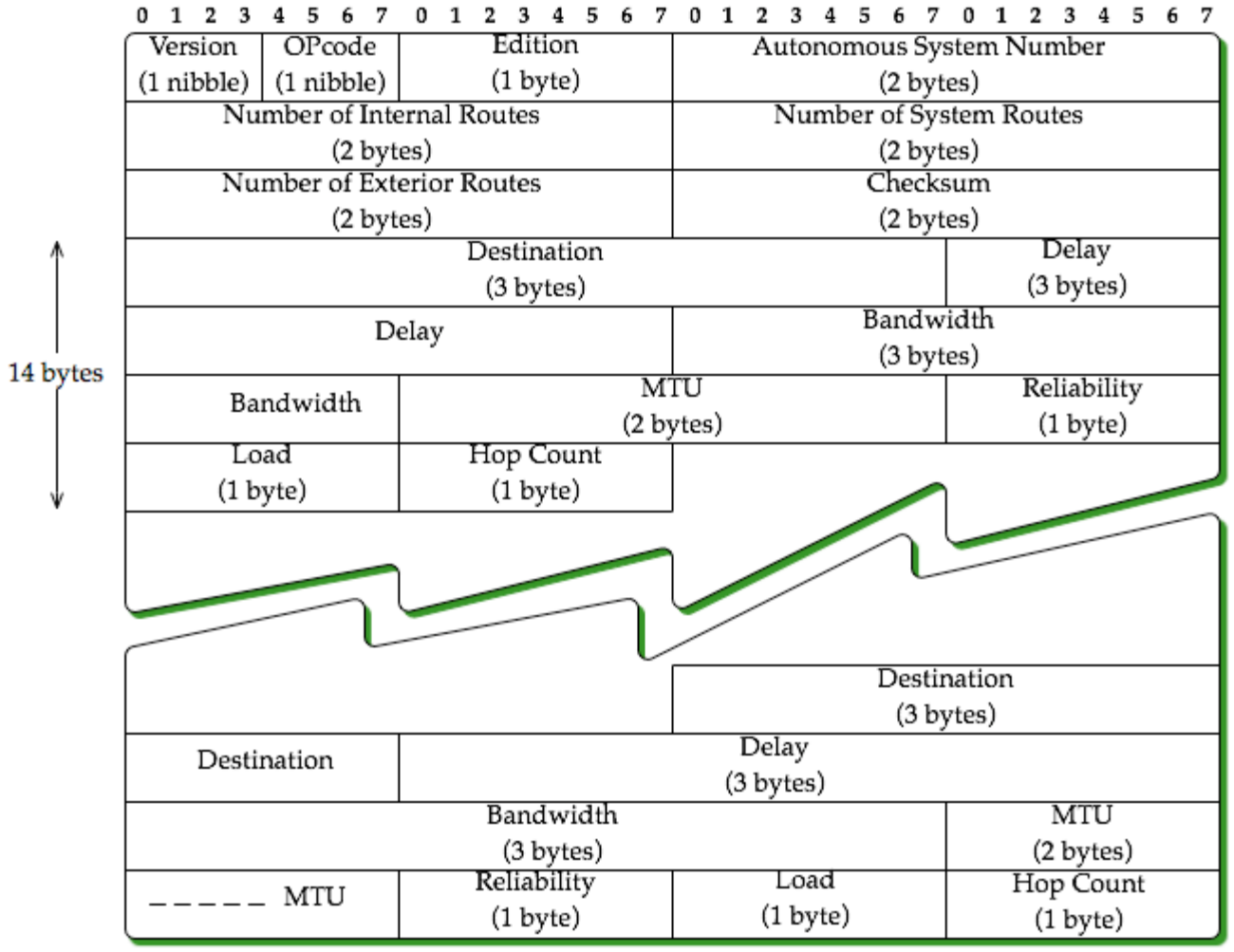
- *Route Tag* (2 bytes): Provided to differentiate internal routes within a RIP routing domain from external routes.
  - For internal routes, this field is set to zero.
  - If a route is obtained from an external routing protocol, then an arbitrary value or preferably the autonomous system number of the external route is included here to differentiate it from internal routes. [more later]
- *Subnet mask* (4 bytes): Routing based on subnet instead of doing classful routing, thus eliminating a major limitation of RIPv1. In particular, variable-length subnet masking (VLSM) may be used.
- *Next hop* (4 bytes): typically, an advertising router is the best next hop from its own view point when it lets its neighbors know about a route; at least, this is the basic assumption.
  - However, in certain unusual circumstances, an advertising router might want to indicate a next hop that is different from itself, such as when two routing domains are connected on the same Ethernet network ([189], [441]).


0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
Command (1 byte)
Version (2 bytes)
Unused (2 bytes)
0xFFFF
Authentication Type (2 bytes)
Authentication
20 bytes
Address Family of net N (2 bytes)
Route Tag (2 types)
IP Address of net 1 (4 bytes)
Subnet Mask to net 1 (4 bytes)
Next Hop (4 bytes)
Hop Distance Metric to net 1 (4 bytes)
20 bytes
Address Family of net N (2 bytes)
Route Tag (2 bytes)
IP Address of net N (4 bytes)
Subnet Mask of net N (4 bytes)
Next Hop (4 bytes)
Hop Distance Metric to net N (4 bytes)

IGRP: Interior Gateway Routing Protocol (developed by Cisco)

- IGRP differs from RIPv1 as follows:
  - IGRP runs directly over IP with protocol type field set to 9.
  - Autonomous system is part of the message fields.
  - Distance vector updates include five different metrics for each route.
  - External routes can be advertised.
  - Allows multiple paths for a route for the purpose of load balancing;
    - Note: requires modification of the Bellman–Ford computation so that instead of a single best path to a destination multiple “almost” equal cost paths can be stored.

IGRP packet format



IGRP: fields

- *Version* (4 bits): This field is set to 1.
- *Opcode* (4 bits): This field is equivalent to the command code in RIP. 1 is a Request and 2 is an Update. In case of a request, only the header is sent; there are no entries.
- *Edition* (1 byte): A counter that is incremented by the sender; this helps prevent a receiving router from using an old update; it essentially plays the role of a timestamp.
- *Autonomous system number* (2 bytes): ID number of an IGRP process.
- *Number of interior routes* (2 bytes): A field to indicate the number of routing entries in an update message that are subnets of a directly connected network.
- *Number of system routes* (2 bytes): This is a counterpart of the number of interior routes; this field is used to indicate the number of route entries that are not directly connected.
- *Number of exterior routes* (2 bytes): The number of route entries that are default networks. This and the previous two fields, the number of interior routes and the number of system routes, together constitute the total number of 14-byte route entries.

IGRP fields [cont'd]

- *Checksum* (2 bytes): This value is calculated on the entire IGRP packet (header + entries).
- *Destination* (3 bytes): This is the destination network for which the distance vector is generated.
  - only 3 bytes instead of the standard 4 bytes for IP addresses.
  - For example, if IP address of a route is 192.168.1.0, entry 192.168.1 is listed in this field.
  - If it is an interior route, the last 3 bytes are listed; for example, if the field lists 16.2.0 for an interior route that is received on interface 172.16.1.254/24, it is meant for the subnet 172.16.2.0.
- *Delay* (3 bytes), *bandwidth* (3 bytes), *reliability* (1 byte), and *load* (1 byte): (see Section 5.5.2 on composite metric)
- *Hop count* (1 byte): A number between 0 and 255 used to indicate the number of hops to the destination.
- *MTU* (2 bytes): The smallest MTU of any link along the route to the destination.

Composite metric

- IGRP introduced a composite metric (EIGRP uses the same one): (5.5.1)

$$C = \begin{cases} \left(K_1 \times B + K_2 \times \frac{B}{256-L} + K_3 \times D\right) \times \left(\frac{K_5}{R+K_4}\right), & \text{if } K_5 \neq 0 \\ K_1 \times B + K_2 \times \frac{B}{256-L} + K_3 \times D, & \text{if } K_5 = 0. \end{cases}$$

  - Has a number of coefficients ("*K*"s), which can be turned off
  - If $K_1 = K_3 = 1$, and $K_2 = K_4 = K_5 = 0$, then

    $C = B + D$

    (How, can you add B and D?)

Digging Deeper

Really, $B = 10^7 / B_{raw}$

$D = D_{raw} / 10$

B_{raw} := Raw link bandwidth expressed in Kbps

D_{raw} : = Raw transmission delay expressed in micro-sec

For an Ethernet link (10Mbps), $B = 10^7/10^4 = 1000$

(For a Fast-Ethernet link, $B = 10^7/10^5 = 100$)

If raw packet transmission delay is 1000 micro-sec,

then D = 100.

Thus, in the case of Ethernet link, "default" composite metric,

$B+D = 1100$

Where does the middle term come from in (5.5.1)

- $L_{raw} = L/256$ (8-bit field), i.e., L_{raw} lies between 0 and 1. Means,
  - L takes a number between 0 and 255.
- Let
  - S be average packet size,
  - λ be arrival rate
  - μ be Service rate, then $\mu = B_{raw}/S$
- M/M/1 delay formula

  $T = 1/(\mu - \lambda)$

Deriving the middle term (5.5.7)

$$T = \frac{1}{\frac{B_{\text{raw}}}{S} - \lambda}.$$

By pulling B_{raw}/S out of the expression in the denominator, we can rewrite it as

$$T = \frac{S}{B_{\text{raw}}} \times \frac{1}{(1 - \frac{S\lambda}{B_{\text{raw}}})}.$$

However, $S\lambda/B_{\text{raw}} = L_{\text{raw}}$ is the raw load. Thus, we arrive at

$$T = \frac{S}{B_{\text{raw}}} \times \frac{1}{(1 - L_{\text{raw}})}.$$

Multiplying the numerator and denominator by 256, we get

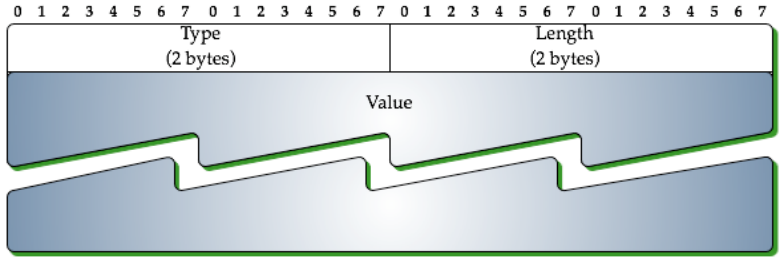
$$T = \frac{S}{B_{\text{raw}}} \times \frac{256}{(256 - 256L_{\text{raw}})}.$$

$$T = \frac{S \times B}{10^7} \times \frac{256}{(256 - L)} = \frac{S \times 256}{10^7} \times \frac{B}{(256 - L)}.$$

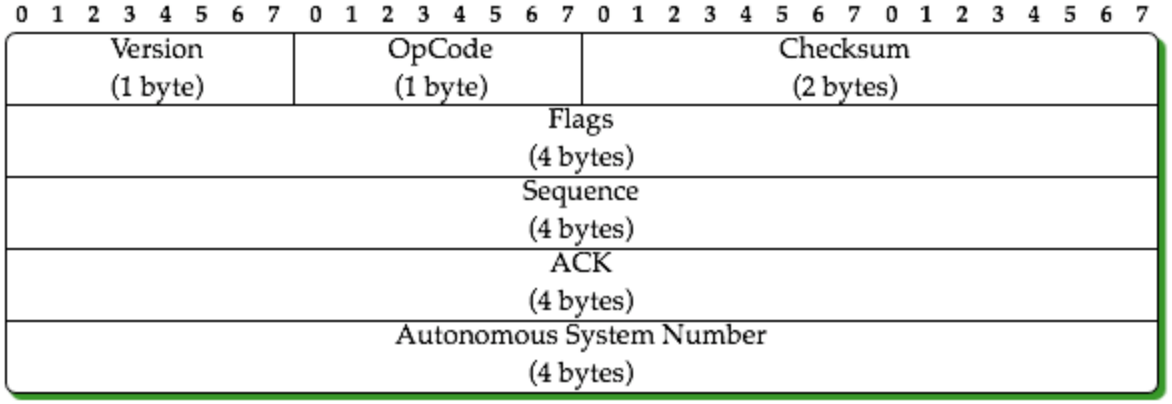
$$C = \begin{cases} (K_1 \times B + K_2 \times \frac{B}{256-L} + K_3 \times D) \times \left(\frac{K_5}{R+K_4}\right), & \text{if } K_5 \neq 0 \\ K_1 \times B + K_2 \times \frac{B}{256-L} + K_3 \times D, & \text{if } K_5 = 0. \end{cases}$$

EIGRP (Enhanced IGRP)

- Main difference from IGRP is that it uses DUAL algorithm (Section 3.3.5), instead of a basic distance vector protocol
- A number of additional information information communicated
- Uses a TLV (Type-Length-Value) encoding scheme
- Common header



Common header



- *Version* (1 byte): 1
- *OpCode* (1 byte): Specify the EIGRP packet type: update, query, reply, and hello. (See Section 3.3.5)
- *Checksum* (2 bytes): Calculated over the entire EIGRP packet.
- *Flags:* 1: indicates a new neighbor relationship.

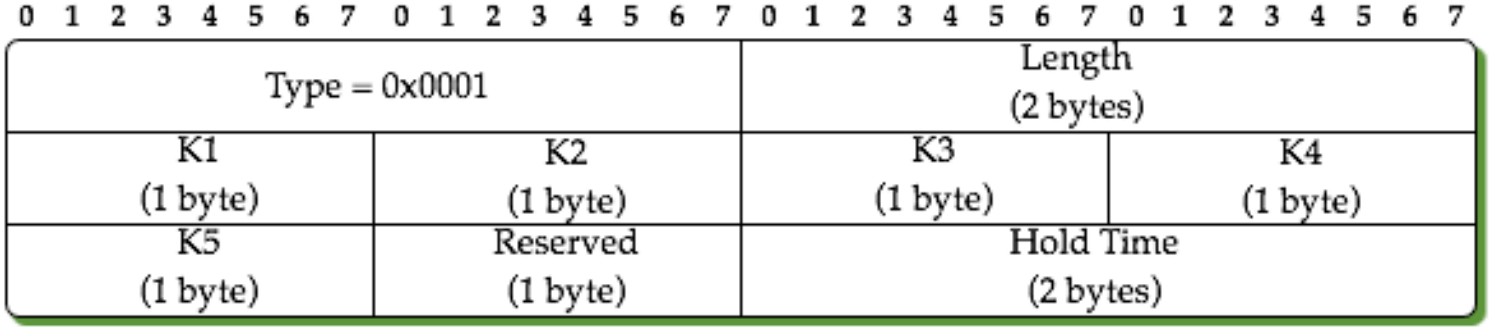
  2: a conditional receive bit for Cisco's propriety multicast algorithm (uses 224.0.0.10)
- *Sequence:* 32-bit sequence number used by the reliable delivery mechanism.
- *ACK:* sequence number from the last heard from neighbor.

  For initial hello packet: set to zero.

  hello packet type with a nonzero ACK: means ACK to initial hello message. [acknowledgment is sent as a unicast message]
- *Autonomous system number:* This identifies the EIGRP domain

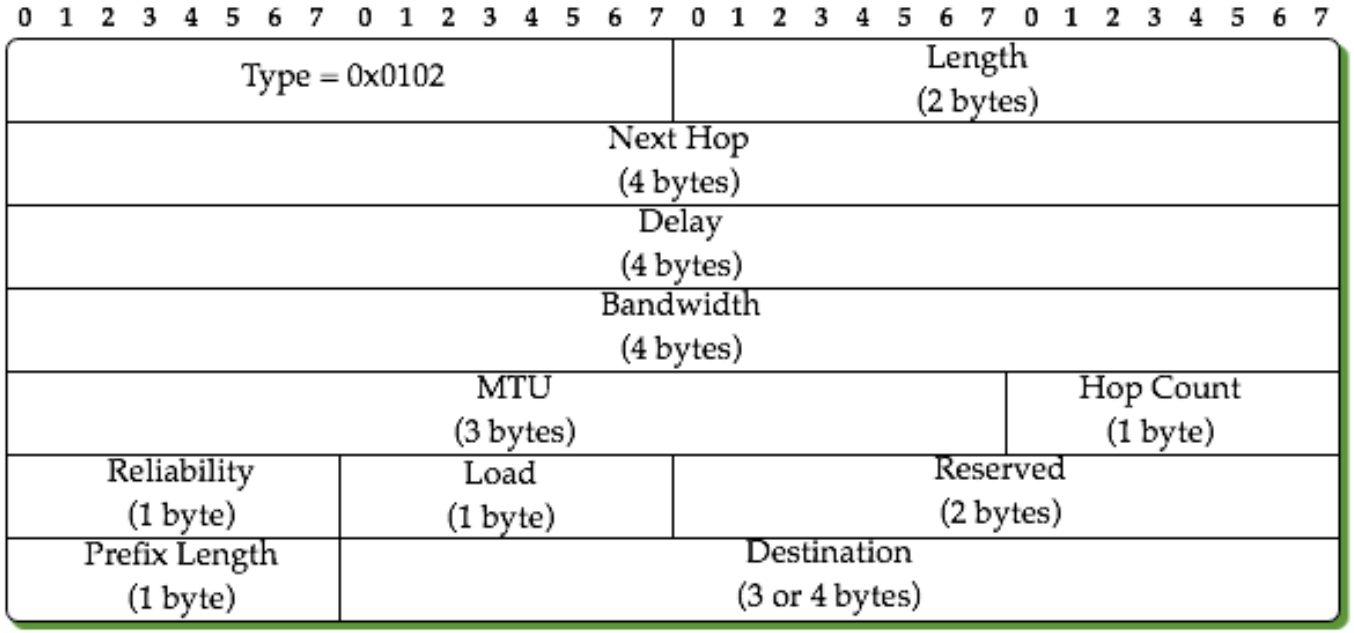
EIGRP metric parameters

- EIGRP allows coefficients used by a router to be communicated to its neighbor (unlike IGRP, although both have the same composite metric)



EIGRP: TLV for communicating DV of an internal route

- Metric components used in composite metric



Summary comparison

TABLE 5.2 Comparison of protocols in the distance vector protocol family.

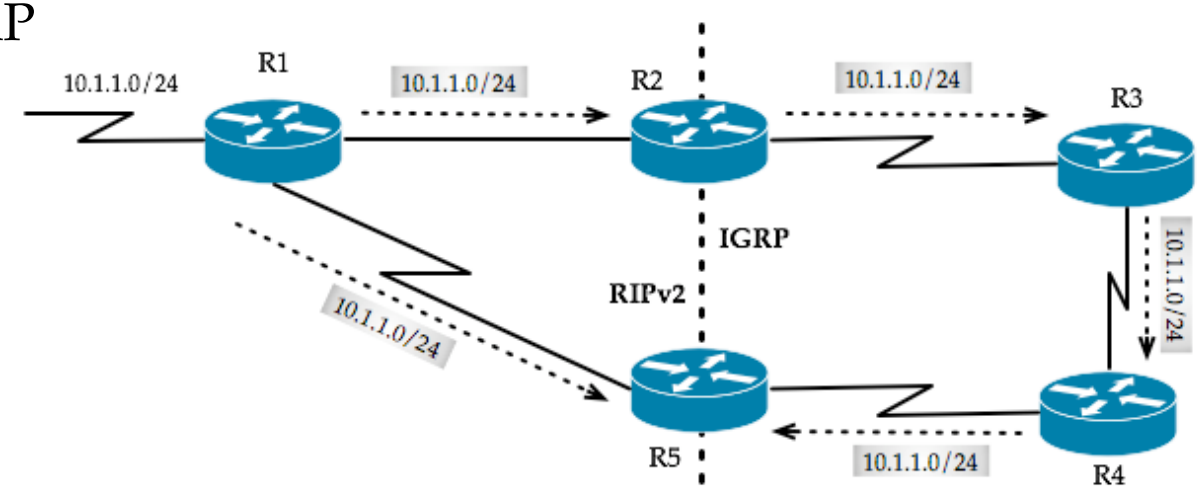
Protocol	RIPv1	RIPv2	IGRP	EIGRP	RIPng
Address Family	IPv4	IPv4	IPv4	IPv4	IPv6
Metric	Hop	Hop	Composite	Composite	Hop
Information Communication	Unreliable, broadcast	unreliable, multicast	Unreliable, multicast	Reliable, multicast	Unreliable, multicast
Routing Computation	Bellman–Ford	Bellman–Ford	Bellman–Ford	Diffusing computation	Bellman–Ford
VLSM/CIDR	No	Yes	No	Yes	v6-based
Remark	Slow convergence; split horizon	Slow convergence; split horizon	Slow convergence; split horizon	Fast, loop-free convergence; chatty protocol	Slow convergence; split horizon

Route Redistribution

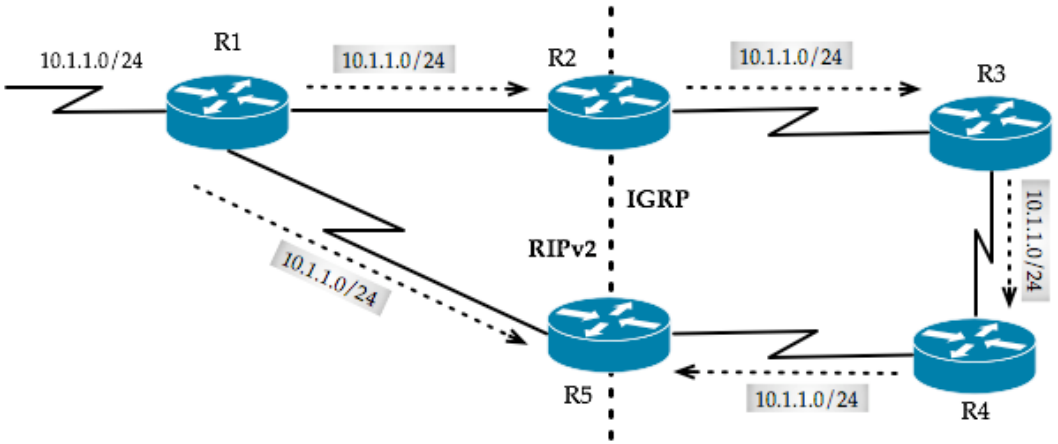
- Consider two different protocols running in adjacent networks,
  - How to inject information from one network to the other
    - Key information is announcement of an IP address block (“route”)
    - The border router must know both protocols!

Route Redistribution [cont'd]

- For letting another protocol know about available routes
  - RIPv2 and IGRP



  - Might be announced through different routers
    - Still convergence/looping problem
      - Note different metrics are used (hop count vs. composite)
      - Use administrative weight?



- Can we avoid looping
- Consider
  - 10.1.1.0/24 ("route") attached to R1 is announced to R2 and R5 (using RIPv2)
  - R2, on learning about this route announces this external route to R3
  - Eventually R5 learns from both RIPv2 side (from R1) and IGRP side (from R4)
    - Better to forward from traffic from R5 to R1 destined for 10.1.1.0/24
    - But this would not happen if you give lower administrative weight to route learned from IGRP compared to RIPv2
    - Split horizon in RIPv2 can avoid looping
    - But, if R3 and R4 fail and come back up again, we can still face problem

Summary

- An overview of IP routing and information typically contained in a routing table
  - Related to netmasking
- Coverage of well known distance vector protocols:
  - RIPv1, RIPv2, IGRP, EIGRP
- Route redistribution