ECE 356/COMPSI 356
Computer Network Architecture

Routing.
Distance-Vector Routing.

Monday September 30th, 2019

Recap

- Last lecture: IP fragmentation, ARP, ICMP
- Material for this lecture: **PD 3.3.1, 3.3.2**

Lecture Outline

- Introduction to routing
- Distance vector routing
- Routing Information Protocol (RIP)

Forwarding and Routing

- There are two parts related to IP packet handling:

1. Forwarding
2. Routing: distributed computation

Static vs. Dynamic Routing

- Two approaches:
  - ➢ Static Routing (Lab 2)
  - ➢ Dynamic Routing
    - Routes are calculated by a routing protocol
    - Graph algorithms
  - ➢ Why do we need a distributed protocol to setup routing tables?

Static Routing

- Manually configure all routes
- Applicable in some cases. E.g.,
  - ➢ If a destination has the same network number as the host, send directly to the destination
  - ➢ Otherwise, send to default router
- Does not deal with failures, implies costs cannot change, does not deal with additions of new nodes or links
  - ➢ Not manageable for large networks

Protocols vs. Algorithms

- Routing protocols establish forwarding tables at routers
- A routing protocol specifies
  - What messages are sent
  - When are they sent
  - How are they handled
- At the heart of any routing protocol is a distributed algorithm that determines the path from a source to a destination

What Distributed Routing Algorithms Common Routing Protocols Use

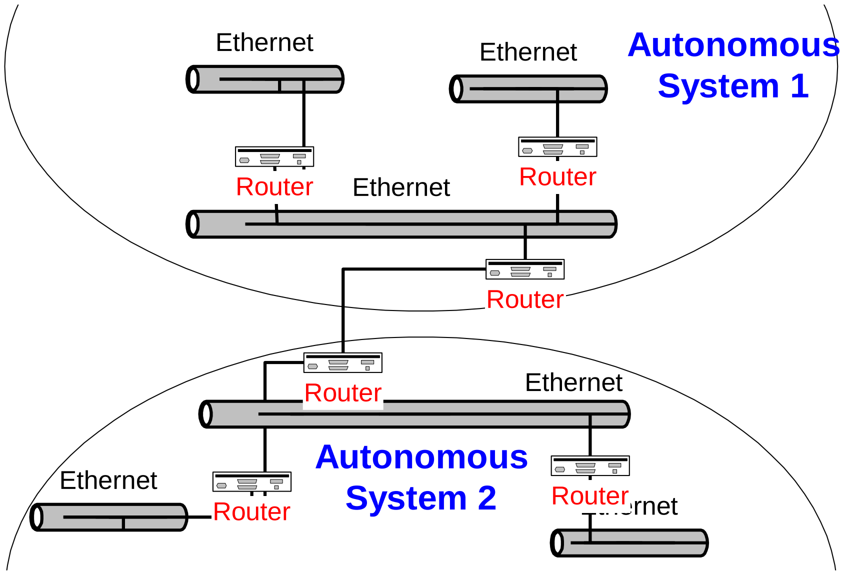
Routing protocol	Distributed algorithm
Routing information protocol (RIP)	Distance vector
Interior Gateway routing protocol (IGRP, CISCO proprietary)	Distance vector
Open shortest path first (OSPF)	Link state
Intermediate System-to-Intermediate System (IS-IS)	Link state
Border gateway protocol (BGP)	Path vector

Intra-domain Routing vs. Inter-domain Routing

- The Internet is a network of networks
- Administrative autonomy
  - Internet = network of networks
  - Each network admin may want to control routing in its own network
- Scale: with 200 million destinations:
  - Cannot store all destinations in routing tables!
  - Routing table exchange would swamp links
  - Solution: using hierarchy to scale

Autonomous Systems (1/2)

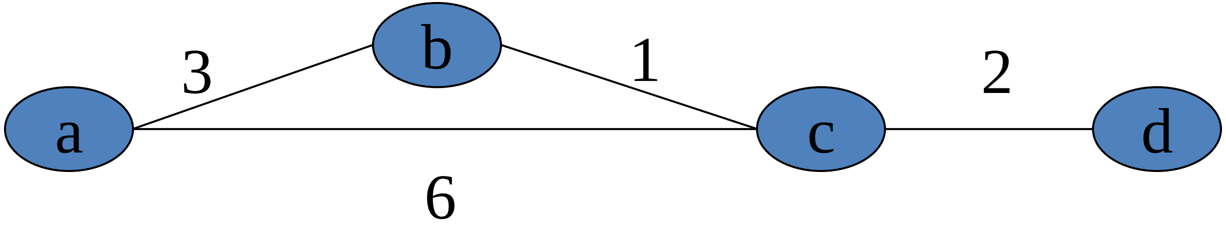
- Aggregate routers into regions, autonomous systems (ASs) or domains
- Routers in the same AS run the same routing protocol
  - “Intra-AS” or intra-domain routing protocol
  - Routers in different AS can run different intra-AS routing protocols



Autonomous Systems (2/2)

- An autonomous system is a region of the Internet that is administered by a single entity
  - Duke's campus network
  - AT&T's backbone network
  - Regional Internet Service Provider (NC regional)
- Intradomain, interdomain routing
- RIP, OSPF, IGRP, and IS-IS are intra-domain routing protocols
- BGP is the only inter-domain routing protocol

Routing Algorithms Compute Shortest Paths in the Network



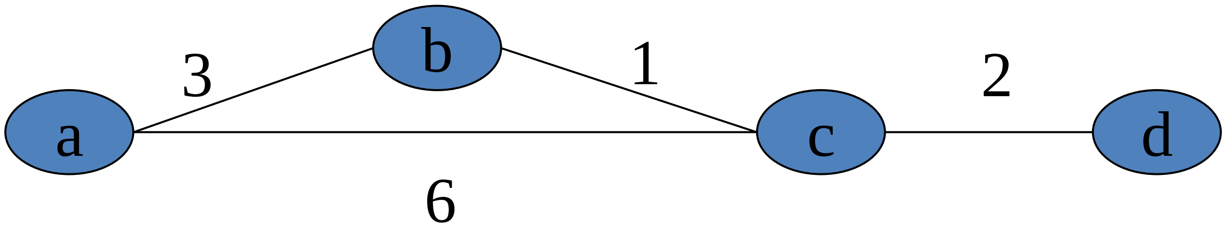
- Shortest path routing algorithms
  - ➢ **Goal**: Given a network where each link is assigned a cost. Find the path with the least cost between two nodes
  - ➢ Shortest path routing is provably loop-free
    - Why?

Lecture Outline

- Introduction to routing
- **Distance vector routing**
- Routing Information Protocol (RIP)

Distance Vector Algorithm: An Introduction

- A decentralized algorithm
  - ➢ Each node has a partial view
    - Neighbors
    - Link costs to neighbors
- *Distance vector*



Distance Vector Algorithm (1/2)

- Path computation is iterative and mutually dependent

1. A router sends its known distances to each destination (distance vector) to its neighbors
2. A router updates the distance to a destination from all its neighbors' distance vectors

Distance Vector Algorithm (2/2)

3. A router sends its updated distance vector to its neighbors
4. The process repeats until all routers' distance vectors do not change
   - *Convergence*

A Router Updates its Distance Vectors using Bellman-Ford Equation

Define

$d_x(y)$:= cost of the least-cost path from x to y

Then

- $d_x(y) = min_v\{c(x,v) + d_v(y)\}$, where *min* is taken over a set *v* of all neighbors of node x

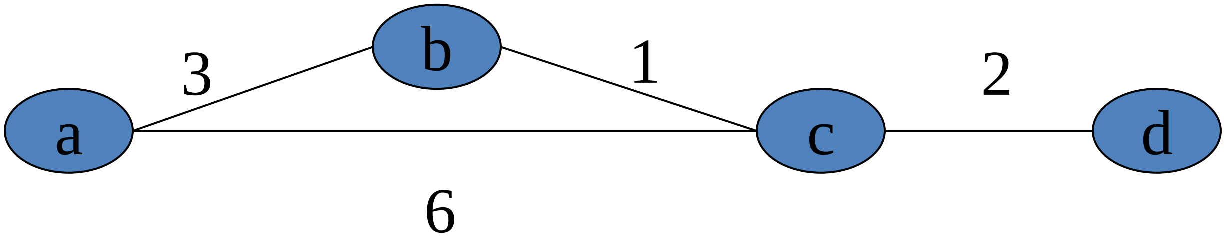
Distance Vector Algorithm: Initialization

- Let $D_x(y)$ be the estimate of least cost from x to y
- Initialization:
  - Each node x knows the cost to each neighbor, $c(x,v)$
  - For each neighbor v of x, $D_x(v) = c(x,v)$
  - $D_x(y)$ to other nodes are initialized as infinity
- Each node x maintains a distance vector (DV):
  - $\mathbf{D}_x = [D_x(y): y \in N]$

Distance Vector Algorithm: Updates

- Each node x sends its distance vector to its neighbors, either periodically, or triggered by a change in its DV
- When a node x receives a new DV estimate from a neighbor v, it updates its own DV using the B-F equation:
  - If $c(x,v) + D_v(y) < D_x(y)$ then
    - $D_x(y) = c(x,v) + D_v(y)$
    - Sets the next hop to reach the destination y to the neighbor v
    - Notify neighbors of the change
- The estimate $D_x(y)$ will converge to the actual least cost $d_x(y)$

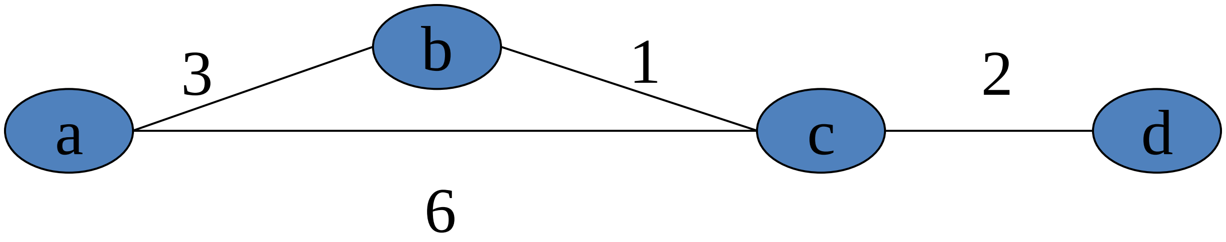
Distance Vector Algorithm: An Example (1/2)



- t = 0
- a = ((a, 0), (b, 3), (c, 6))
- b = ((a, 3), (b, 0), (c,1))
- c = ((a, 6), (b, 1), (c, 0) (d, 2))
- d = ((c, 2), (d, 0))

- t = 1
- a = ((a, 0), (b, 3), (c, 4), (d, 8))
- b = ((a, 3), (b, 0), (c,1), (d, 3))
- c = ((a, 4), (b, 1), (c, 0), (d, 2))
- d = ((a, 8), (b, 3), (c, 2), (d,0))

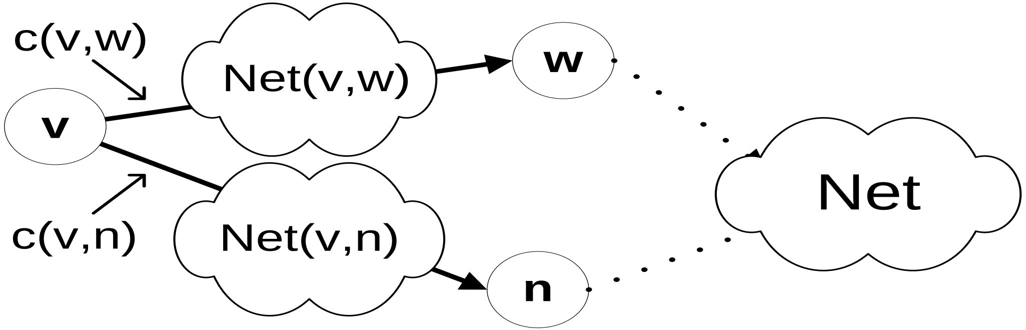
Distance Vector Algorithm: An Example (2/2)



- t = 1
- a = ((a, 0), (b, 3), (c, 4), (d, 8))
- b = ((a, 3), (b, 0), (c,1), (d, 3))
- c = ((a, 4), (b, 1), (c, 0), (d, 2))
- d = ((a, 8), (b, 3), (c, 2), (d,0))

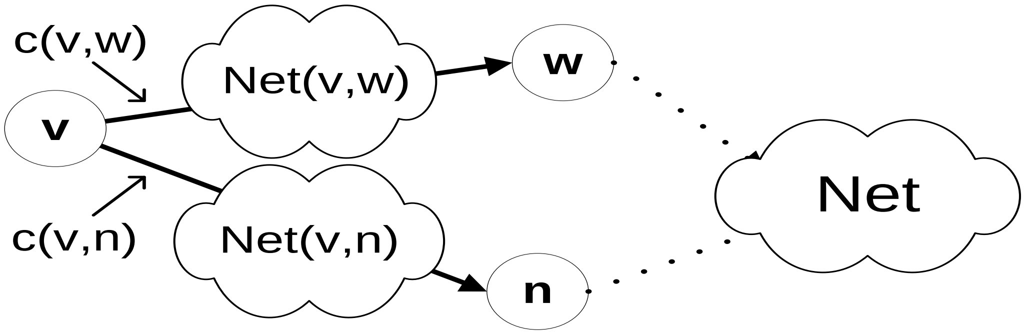
- t = 2
- a = ((a, 0), (b, 3), (c, 4), (d, 6))
- b = ((a, 3), (b, 0), (c,1), (d, 3))
- c = ((a, 4), (b, 1), (c, 0), (d, 2))
- d = ((a, 6), (b, 3), (c, 2), (d,0))

Mapping an Abstract Graph to the Physical Network (1/2)



- Nodes (e.g., v, w, n) are routers, identified by IP addresses, e.g. 10.0.0.1
- Nodes are connected by either a directed link or a broadcast link (Ethernet)

Mapping an Abstract Graph to the Physical Network (2/2)

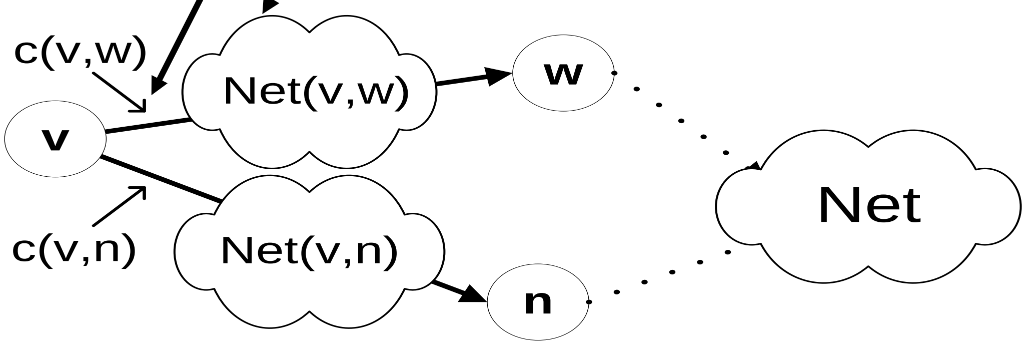


- Destinations are IP networks, represented by the network prefixes, e.g., 10.0.0.0/16
  - Net(v,n) is the network directly connected to router v and n
- Costs (e.g. c(v,n)) are associated with network interfaces

Distance Vector Routing Protocol: Routing Table

c(v,w): cost to transmit on the interface to network Net(v,w)

Net(v,w): Network address of the network between v and w



RoutingTable of node v

Dest	via (next hop)	cost
Net	n	D(v,Net)

- D(v,Net) is v's cost to Net

Distance Vector Routing Protocol: Messages

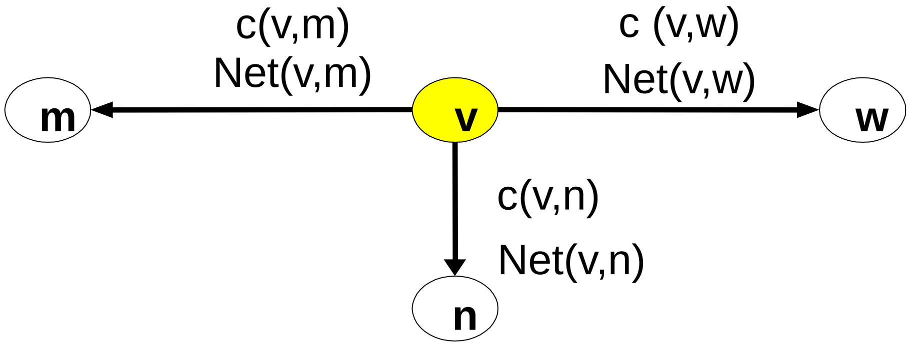
- Nodes send messages to their neighbors which contain distance
- A message has the format: [Net , D(v,Net)] means *"My cost to go to Net is D (v,Net) "*

RoutingTable of node v

Dest	via (next hop)	cost
Net	n	D(v,Net)

Initiating Routing Table (1/3)

- Suppose a new node v becomes active
- The cost to access directly connected networks is zero



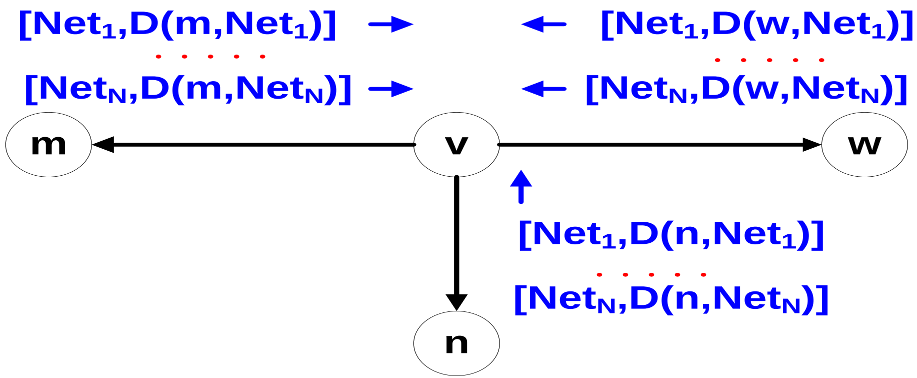
Routing Table

Dest	via (next hop)	cost
Net(v,m)	-	0
Net(v,w)	-	0
Net(v,n)	-	0

D (v, Net(v,m)) = D (v, Net(v,w)) = D (v, Net(v,n)) = 0

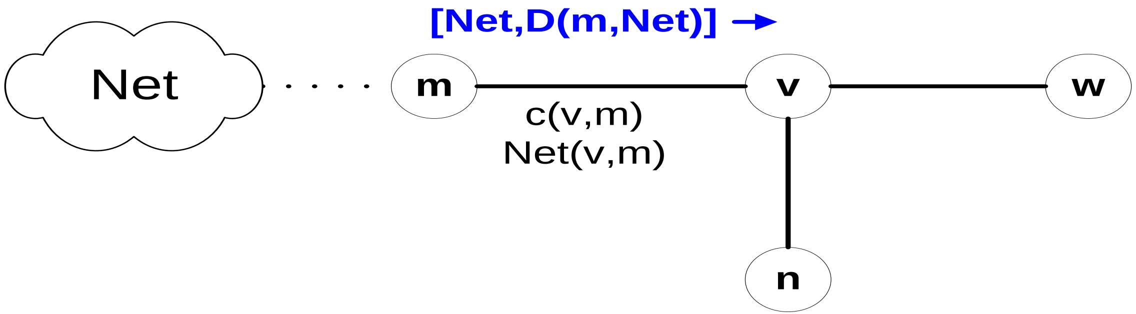
Initiating Routing Table (2/3)

- Node v receives the routing tables from other nodes and builds up its routing table



Updating Routing Tables (3/3)

- Suppose node v receives a message from node m: **[Net,D(m,Net)]**



Node v updates its routing table and sends out further messages if the update reduces the cost of a route:

if (D(m,Net) + c (v,m) < D (v,Net)) {
D^{new} (v,Net) := D (m,Net) + c (v,m);
Update routing table;
send message [Net, D^{new} (v,Net)] to all neighbors
}

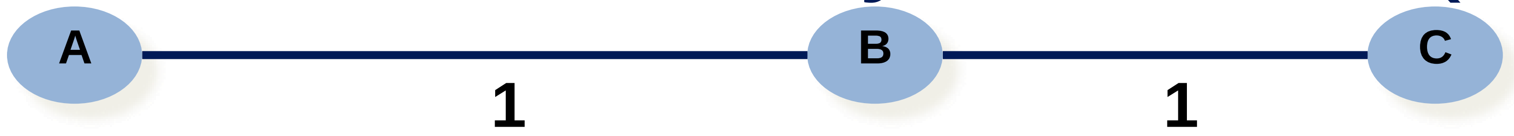
Characteristics of Distance Vector Routing Protocols (1/2)

- **Periodic updates:** Updates to the routing tables are sent at the end of a certain time period. A typical value is 30 seconds
- **Triggered updates:** If a metric changes on a link, a router immediately sends out an update without waiting for the end of the update period

Characteristics of Distance Vector Routing Protocols (2/2)

- **Full routing table update:** Most distance vector routing protocol send their neighbors the entire routing table (not only entries which change)
- **Route invalidation timers:** Routing table entries are invalid if they are not refreshed. A typical approach is to invalidate an entry if no update is received after 3-6 update periods

The Count-to-Infinity Problem (1/2)



A's Routing Table

to	via (next hop)	cost
C	B	2

B's Routing Table

to	via (next hop)	cost
C	C	1

now link B-C goes down

C	B	2

C	-	∞

C	2

→ ←

C	∞

C	-	∞

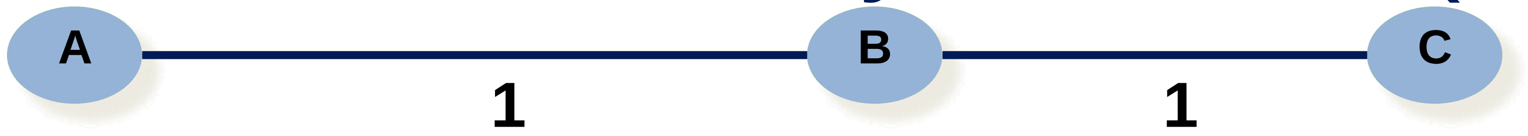
C	A	3

C	∞

→ ←

C	3

The Count-to-Infinity Problem (2/2)



A's Routing Table				B's Routing Table		
C	B	2		C	-	∞
	C	2	→ ←	C	∞	
C	-	∞		C	A	3
	C	∞	→ ←	C	3	
C	B	4		C	-	∞
	C	4	→ ←	C	∞	

Count-to-Infinity Problem: The Cause

- The reason for the count-to-infinity problem is that each node only has a "next-hop-view"
  - ➢ Would not happen with global knowledge
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B

Solutions to Count-to-Infinity Problem

- Always advertise the entire path in an update message to avoid loops (Path vectors)
  - BGP uses this solution

Remedies to Count-to-Infinity Problem

- Never advertise the cost to a neighbor if this neighbor is the next hop on the current path (“*split horizon*”)
  - ➢ Example: A would not send the first routing update to B, since B is the next hop on A’s current route to C
  - ➢ *Split horizon with poison reverse*
    - Sends to the next hop neighbor an invalid route (C, ∞)
  - ➢ Only solve the problem if routing loops involve only two nodes
- Have a small infinity (e.g., 16) so that routing messages will not bounce forever

Lecture Outline

- Introduction to routing
- Distance vector routing
- **Routing Information Protocol (RIP)**
  - Lab 3 is about RIP

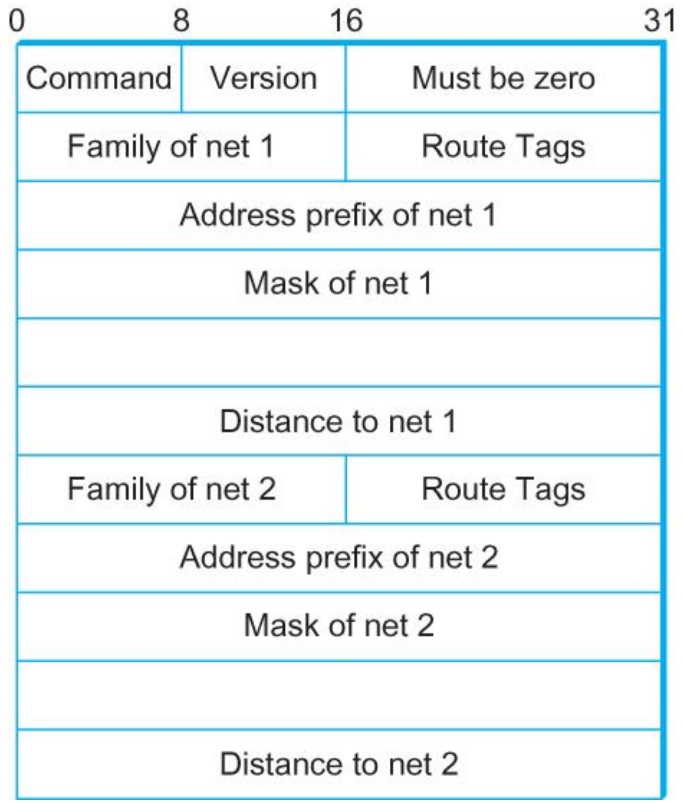
Routing Information Protocol (RIP)

- A simple intra-domain protocol
- Straightforward implementation of Distance Vector Routing
- Each router advertises its distance vector every 30 seconds (or whenever its routing table changes) to all adjacent routers
- Always uses 1 as link metric
- Maximum hop count is 15, with “16” equal to “∞”
- Routes are timed out (set to 16) after 3 minutes if they are not updated

RIPv2 Packet Format

IP header	**UDP header**	**RIP Message**

- Runs on top of UDP
- Supports multiple address families (not just IP)
- Up to 25 route entries per message



RIP Problems

- RIP takes a long time to stabilize
  - Even for a small network, it takes several minutes until the routing tables have settled after a change
- RIP has all the problems of distance vector algorithms, e.g., count-to-Infinity
  - RIP uses split horizon to avoid count-to-infinity
- The maximum path in RIP is 15 hops

Lecture Summary

- Introduction to routing
- Distance vector routing
- Routing Information Protocol (RIP)

Next Lectures

- Next lecture: Link state routing
- Lecture after next: Routing between domains