COMP9332

Network Routing & Switching

RIP

www.cse.unsw.edu.au/~cs9332

Review (1)

- Routing Approaches
  - Flooding
  - Source routing
  - Forwarding Table
  - Spanning tree
- Metrics
- Shortest path spanning tree
- Other types of routing
  - Multipath
  - Multicast

Review (1)

- IP addressing
  - IP uses hierarchical addressing
    - » Network id, subnet id, hostid
    - » Network prefix, hostid
- Routing
  - The aim is to find a route from the source to the destination
  - For static routing, administrator sets up routing table manually

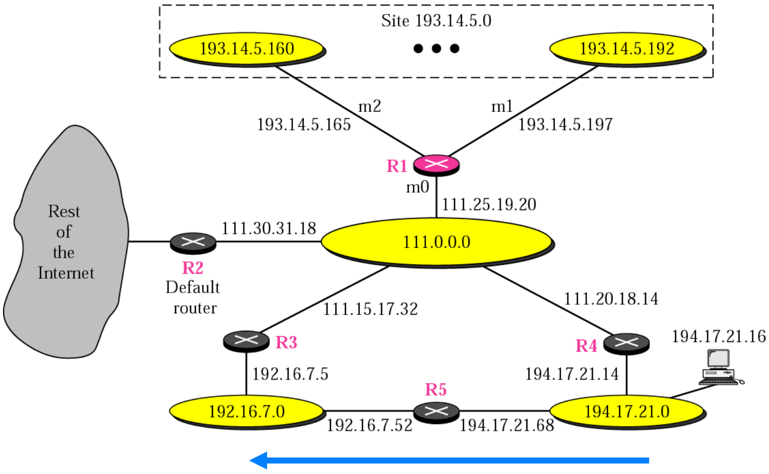
Review (2)

- Routing table mostly specifies the next hop rather than the complete route
  - Each row of the table consists of
    - » Destination network (Network prefix and prefix length)
    - » Next Hop to the destination network
- Because of hierarchical addressing
  - Routing table contains addresses of networks and rarely those of specific hosts
  - This reduces the size of the routing tables

Problems with static routing table

- The administrative burden for static routing table is huge for large networks
- Static routing table cannot cope with link or router failure. For example,
  - If 194.17.21.0 in "Our Internet" is statically configured to use R5 to reach 192.16.7.0
  - Failure of R5 means hosts in 192.16.7.0 become unreachable even an alternative route exists
    [next slide]

"Our Internet"



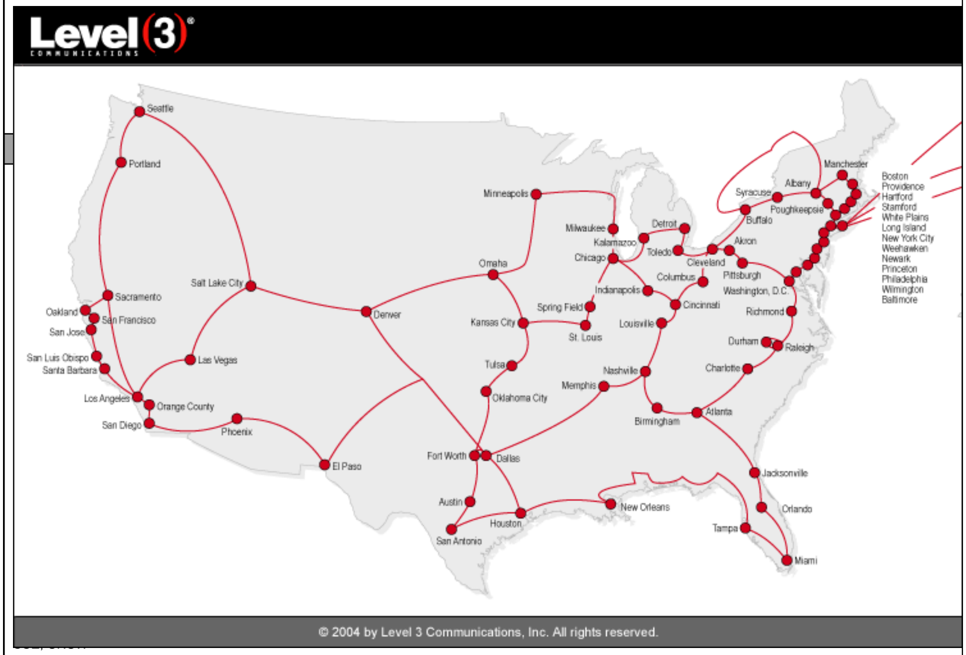
Static route to 192.16.7.0 from 194.17.21.0

Dynamic Routing Table

- Routers build their routing table by exchanging information with each other
- Table entries need to be updated when the network condition changes e.g. a link has failed, a link has come up etc.
- Routing algorithms and protocols are used to update routing tables automatically

What is needed to build a routing table automatically?

- Given the network map on the next page, how can you find a route from Seattle to Miami?
- You need
  - Some network information
  - Some way to choose a route from the multiple routes available



Routing protocols

- Define how routers exchange network information
  - What type of information
  - The format of information exchange
  - When to exchange
  - Which router to exchange information with

Routing Algorithms

- Aim: To choose or compute a route based on the available network information
- A routing algorithm is defined by
  - The type of network information exchanges
  - Which router to exchange information with
  - Method to compute the routes

Routing algorithms and protocols

- Routing algorithm versus routing protocol
  - Routing algorithm is generic
  - Routing protocol is a specific implementation of a routing algorithm
- Two main classes of routing algorithms
  - Distance vector
  - Link state
- These two routing algorithms are used very often
  - *Make sure you know them well!*

This lecture

- Routing algorithm
  - Distance vector
- Routing protocols
  - RIP
    - Based on distance vector
- We begin by looking at the abstract model that routing algorithms use

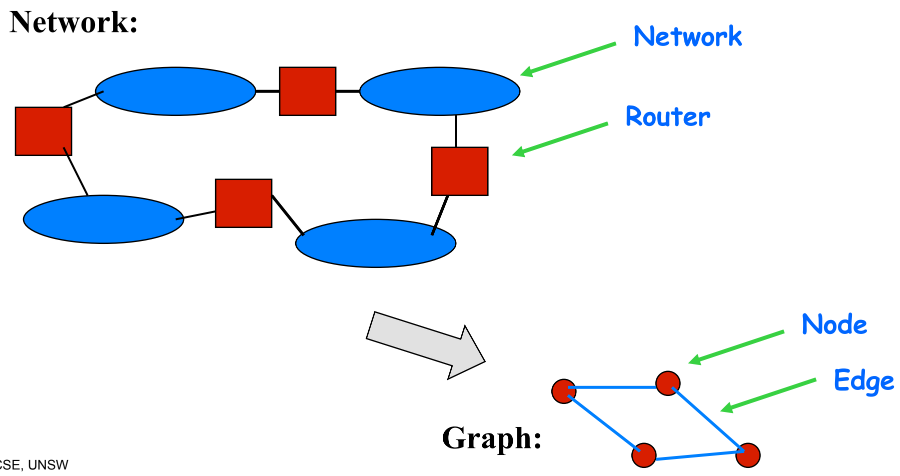
CSE, UNSW

Network as a graph (1)

- In order to make routing algorithms general, a network is abstracted as a graph
- A graph is a mathematical abstraction
- A graph is specified by
  - A set of nodes
  - A set of edges (links)

CSE, UNSW

Network as a graph: example



Network as a graph (2)

- Conversion principles
  - Routers become nodes of the graph
  - Two nodes in the graph are connected by an edge if the corresponding routers are connected by a network or network link

Routing algorithm basics

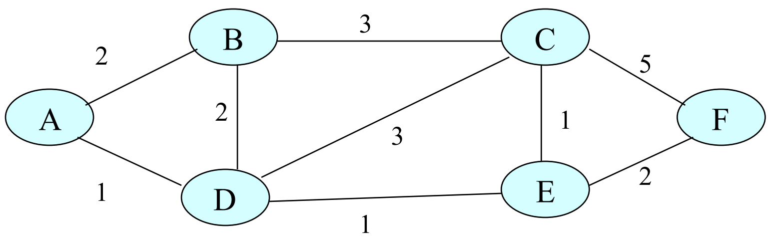
- Each edge of the graph is assigned a cost
  - Example 1: Unit cost per edge
  - Example 2: The cost of an edge expresses the desirability of using the corresponding network link e.g. a high bandwidth link has a low cost
    - » See the example on the next slide
- The cost of a route is the sum of costs of the constituent links
- The aim of both distance vector and link state routing algorithms is to find the least cost path or shortest path

Example: Cost of edges and routes

Two possible routes from A to E

Cost of route A → B → C → E is 2+3+1 = 6

Cost of route A → D → E is 1 + 1 = 2



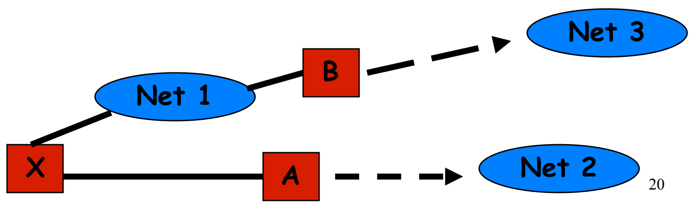
Distance vector routing algorithm

- Messages exchanged between routers have two components:
  - destination network (*vector*)
  - cost to destination (*distance*)
- Also known as *Bellman-Ford* Algorithm
  - Invented in the 1960's

Distance vector: routing table

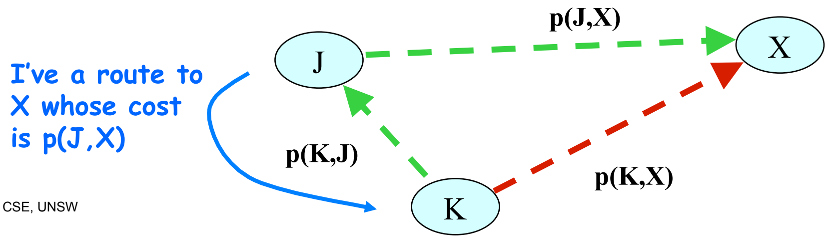
- Each router maintains a routing table
  - Example: The routing table for a router X

Destination	Cost	Next Hop
Net 1	0	Direct
Net 2	5	Router A
Net 3	3	Router B



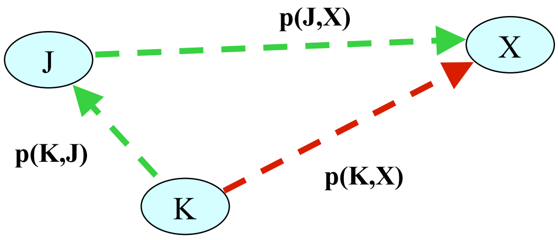
The secret behind distance vector (1)

- **Given:**
  - Router K has an existing route to network X with cost p(K,X)
  - A neighbouring router J tells K that it has a route to X with cost p(J,X)
  - Cost between J and K is p(K,J)
- **Q: Should K use the existing route or use the one via J?**



The secret behind distance vector (2)

- **K has two choices**
  - Existing route with cost p(K,X)
  - Alternative route (via J) with cost p(K,J) + p(J,X)
- **Distance vector routing algorithm**
  - If p(K,J)+p(J,X) < p(K,X): use the route via J
  - Otherwise, use the existing route

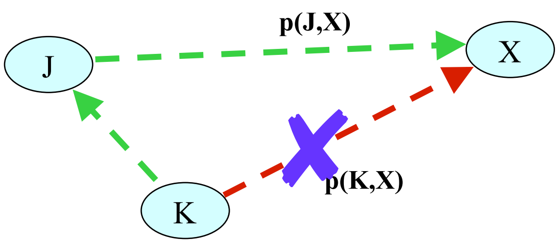


Distance vector: route update (1)

- Periodically, each router sends a copy of its table (destination, cost columns only) to *directly connected* routers
- When router *K* receives table from a neighbouring router *J*, *K* updates its table if:
  - J knows a shorter route for a given destination
  - J knows a destination K didn't know about
  - K currently routes to a destination through J and J's cost to that destination has changed

Distance vector: route update (2)

- If *K* updates or adds an entry in response to *J*'s message,
  - It assigns the Next Hop as Router *J*
  - It updates the cost. If X is the destination, then cost to X = p(K,J) + p(J,X)



Exercise: Route update in distance vector

Existing routing table for router K

Destination	Cost	Next Hop
Net 1	0	Direct
Net 2	0	Direct
Net 4	8	Router L
Net 17	5	Router M
Net 24	6	Router J
Net 30	2	Router Q
Net 42	2	Router J

Routing table from neighbouring router J:

Destination	Cost
Net 1	2
Net 4	3
Net 17	6
Net 21	4
Net 24	5
Net 30	10
Net 42	3

- Router K receives the routing table from neighbouring router J (showed on the right). If the cost between them is 1, what is the routing table for K after the update?

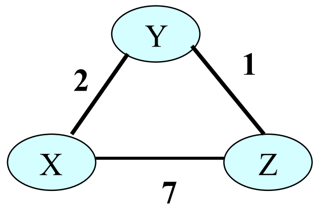
Solution

Destination	Cost	Next Hop
Net 1	0	Direct
Net 2	0	Direct
Net 4 (*)	4	Router J
Net 17	5	Router M
Net 21 (*)	5	Router J
Net 24	6	Router J
Net 30	2	Router Q
Net 42 (*)	4	Router J

The entries marked with a red asterisk (*) are updated due to the information from router J

Distance Vector: Start up (1)

- At bootup each router initialises its routing table with directly connected network information
- Example: The following shows a network with 3 nodes. The cost of an edge is showed next to the edge.



The initial routing table for X is:

Destination	Cost	Next Hop
Y	2	Direct
Z	7	Direct

Distance Vector: Start up (2)

- After initialising their routing tables, the nodes start exchanging their routing tables with their neighbouring nodes
- When a node receives a routing table, it updates its routing table according to the distance vector update rules
- Exercise: For the 3-node network on the previous slide, fill in the routing tables on the next page
  1. What are the initial routing tables for Y and Z?
  2. Assuming Y sends its routing table to X and Z, what are their routing tables after the update?

The initial routing tables:

Routing table for X		
Destination	Cost	Next Hop
Y	2	Direct
Z	7	Direct

Routing table for Y		
Destination	Cost	Next Hop

Routing table for Z		
Destination	Cost	Next Hop

Y sends its routing table to its neigbouring nodes and they update their routing tables. After update, the routing tables are:

Routing table for X		
Destination	Cost	Next Hop

Routing table for Y		
Destination	Cost	Next Hop

Routing table for Z		
Destination	Cost	Next Hop

Solution:
The initial routing tables:

Routing table for X		
Destination	Cost	Next Hop
Y	2	Direct
Z	7	Direct

Routing table for Y		
Destination	Cost	Next Hop
X	2	Direct
Z	1	Direct

Routing table for Z		
Destination	Cost	Next Hop
X	7	Direct
Y	1	Direct

Solution: The routing tables after the update are:
Note: * indicates an entry has been updated.

Routing table for X		
Destination	Cost	Next Hop
Y	2	Direct
Z	3 *	Y *

Routing table for Y		
Destination	Cost	Next Hop
X	2	Direct
Z	1	Direct

Routing table for Z		
Destination	Cost	Next Hop
X	3 *	Y *
Y	1	Direct

Distance Vector: Start up (3)

- After this update, X sends its routing table to its neighbouring nodes
  - The routing tables before and after this update is sent is showed on the next slide
- The aim of distance vector routing algorithm is to find the least path route, the update is not causing any changes because the least cost routes have been found

The routing tables before X sends its update:

Routing table for X		
Destination	Cost	Next Hop
Y	2	Direct
Z	3	Y

Routing table for Y		
Destination	Cost	Next Hop
X	2	Direct
Z	1	Direct

Routing table for Z		
Destination	Cost	Next Hop
X	3	Y
Y	1	Direct

The routing tables after X sends its update:
In fact, the tables remain the same.

Routing table for X		
Destination	Cost	Next Hop
Y	2	Direct
Z	3	Y

Routing table for Y		
Destination	Cost	Next Hop
X	2	Direct
Z	1	Direct

Routing table for Z		
Destination	Cost	Next Hop
X	3	Y
Y	1	Direct

Distance Vector: Start up (4)

- Finally, Z sends its routing tables to its neighbouring nodes
  - Their neighbouring nodes updates their routing table
  - The routing table remains the same as before
    - » Because the routing tables already consist of the least cost routes

Operation of Distance Vector

- Distributed - no global view
- Asynchronous - no lock-step updates
- Iterative - several updates until converged

Changes in Distance Vector Tables

The following events may cause a change in the table

- Change of cost of an attached link
- Receipt of an update message from a neighbour

Intermission

- We talked about
  - Routing algorithm: generic
  - Routing protocol: specific implementation
  - A routing algorithm: distance vector
- Next: RIP - a routing protocol which uses distance vector, but before that
- We see how Internet routing protocols are organised

Internet routing protocols

Three standard Internet routing protocols:

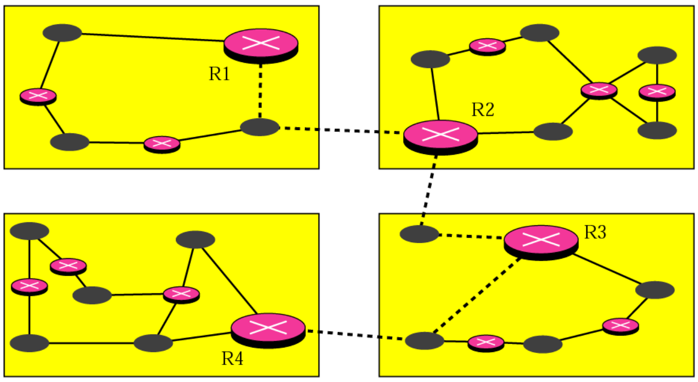
Routing protocol	Based on routing algorithm	IGP/EGP
Routing Information Protocol (RIP)	Distance vector	IGP
Open shortest path first (OSPF)	Link state	IGP
Border Gateway Protocol (BGP)	Path vector	EGP

Note: there are also propriety routing protocols

Organisation of the Internet

- Internet consists of networks interconnected with routers
- Internet is also organised into Autonomous Systems (AS) [Illustration – next slide]
  - An AS consists of multiple networks and routers under one single administration
  - An AS can belong to an organisation, a university etc.
  - An AS is identified by a 16-bit public or private Autonomous System Number
- Three levels of hierarchy: Hosts, Networks, AS
  - Hierarchy helps to deal with scalability

Autonomous system (AS) - an illustration



Interior and exterior gateway routing protocols

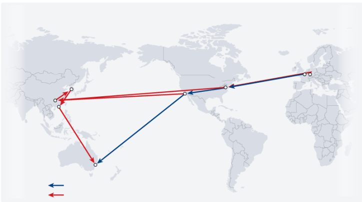
- Two classes of routing protocol
  - Interior gateway protocol (IGP)
    - » Responsible for routing within an AS
  - Exterior gateway protocol (EGP)
    - » Responsible for inter-AS routing

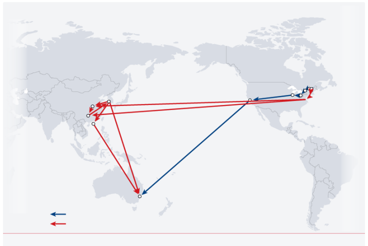
Routing protocol design issues

- Does it work correctly?
- Reliability
  - How to cope with failures?
    - » Lost of routing control packets
    - » Link or router failures
- Performance
  - How much routing control traffic is introduced into the network?
    - » Routing control packets are considered as overheads
  - Size of the routing table

Routing protocol design issues (cont'd)

- Scalability
  - Can we maintain the performance with growing number of nodes, routers, networks
- Security
  - Can people maliciously introduce false routing information in the system?





Routing Information Protocol (RIP)

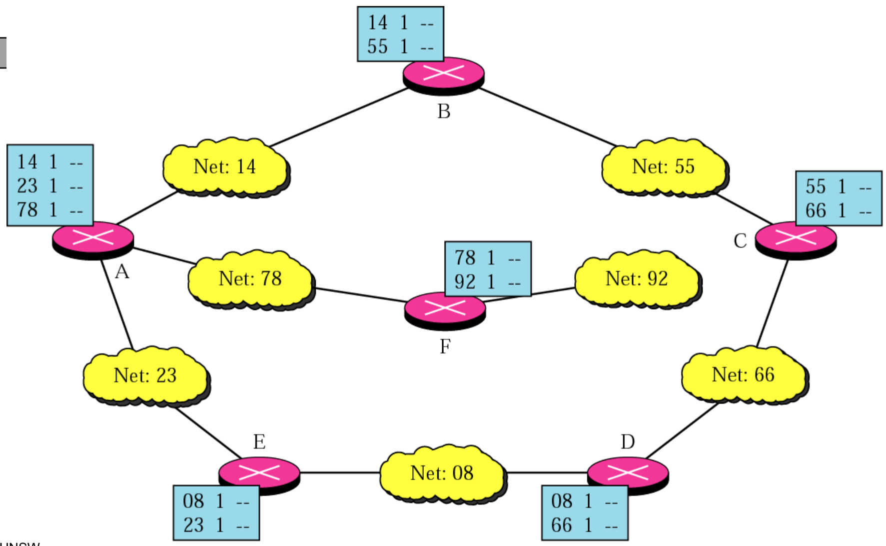
- Implements distance vector routing
  - Each router shares routing information (destination network, hop count to destination) with its neighbours
  - Sharing at regular interval
- Versions: RIP1, RIP2
- Unix RIP implementation
  - routed: Short for "route daemon", pronounced "route-d" (supprts only RIP1)
  - gated: version 2 supports only RIP1, but version 3 supports both RIP1 and RIP2
- RIP messages are sent over UDP (port 520)
  - Some updates may be lost (*periodic updates* to address the problem)

Cost in RIP

- **In RIP, the cost is related to the number of router hops**
  - **Cost in RIP is between 0 and 16**
  - **A cost 16 means infinity, i.e. the network is unreachable**
- **There are two conventions**
  - **Convention 1:**
    - » **For directly connected network, cost = 0**
    - » **To reach a destination via 1 router, cost = 1**
    - » **To reach a destination via 2 routers, cost = 2**
  - **Convention 2:**
    - » **For directly connected network, cost = 1**
    - » **To reach a destination via *n* routers, cost = n + 1**

Initial routing tables in a small autonomous system

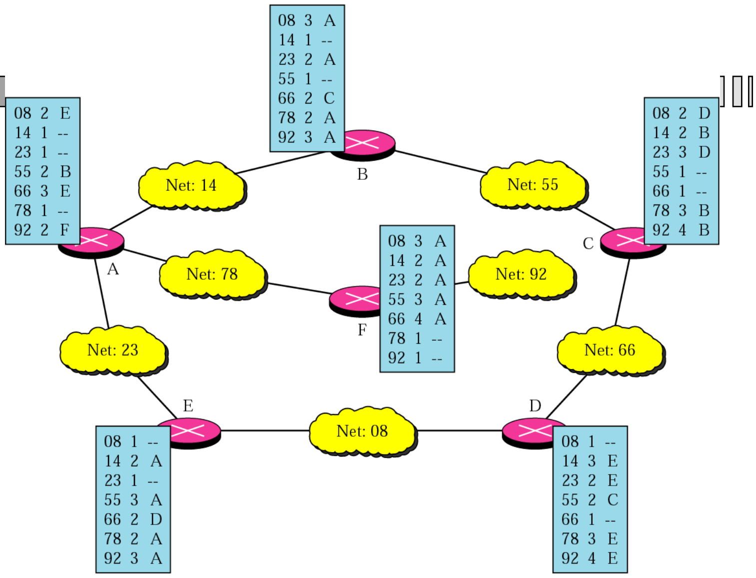
(assume *one hop* to directly connected network)



CSE, UNSW

45

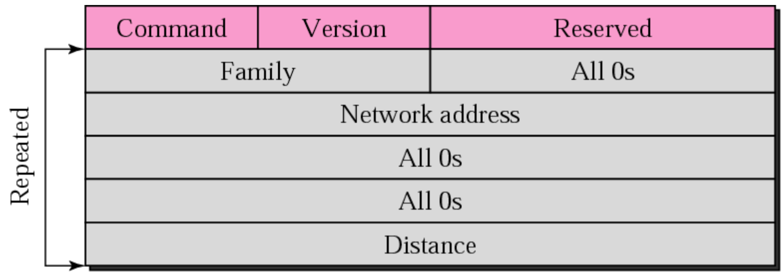
Final routing tables for the previous figure



CSE, UNSW

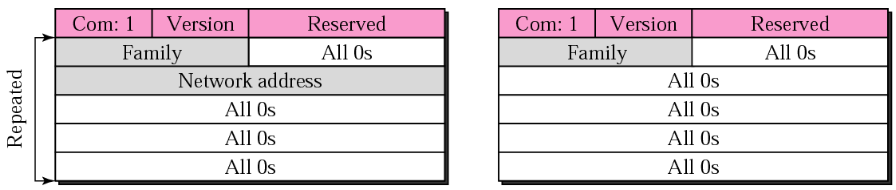
46

RIPv1 message format



Command:
1 = Request
2 = Response

RIPv1 Request message



a. Request for some   b. Request for all

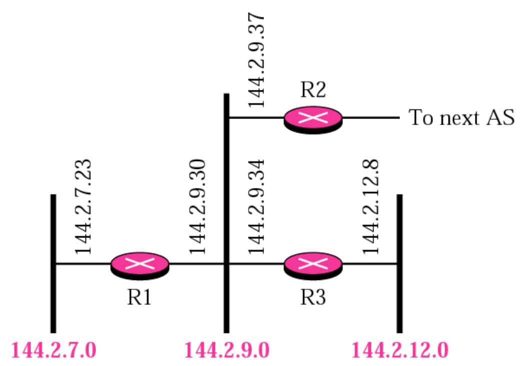
Request message is used by a router that has just come up.

Response message

- As replies to request messages
- For periodic distance vector announcements

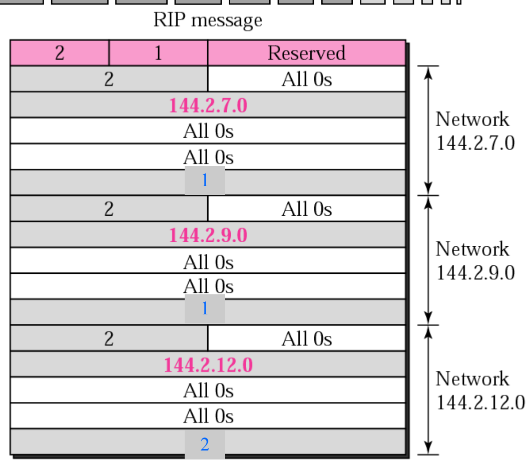
Example – response message

- What is the periodic response sent by router R1?



Example (cont'd)

- R1 advertises three networks
  - 144.2.7.0
  - 144.2.9.0
  - 144.2.12.0
- Note: we used a cost of 1 for directly connected networks



RIP timers (1)

- RIP uses 3 timers to support its operations
- Periodic timer
  - Controls the regular update interval
  - Nominal value 30s but in practice a random number between 25-35s to avoid synchronization
  - Not affected by triggered updates

RIP Timers (2)

- Expiration Timer
  - every route (table entry) has one
  - reset each time update is received for the route
  - expires if not updated in 180 seconds, assign hop count 16 (becomes invalid route)
  - invalid route is advertised with hop count of 16 (other routers learn that this route is invalid)

RIP Timers (3)

- Garbage Collection Timer
  - objective is to ultimately expunge invalid routes from the routing tables
  - is set to 120 seconds when expiration timer expires
  - during this 120 seconds, the router will continue to advertise the invalid route with hop count of 16

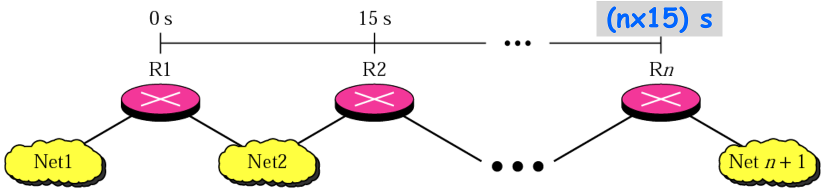
Exercise: RTP timers

- A routing table has 20 entries. It has not received updates on five routes for 200s. How many timers are running at this time?

Solution

- The following timers are running
  - # Periodic timer = 1
  - # Expiration timer = 20 - 5 = 15
  - # Garbage collection timer = 5

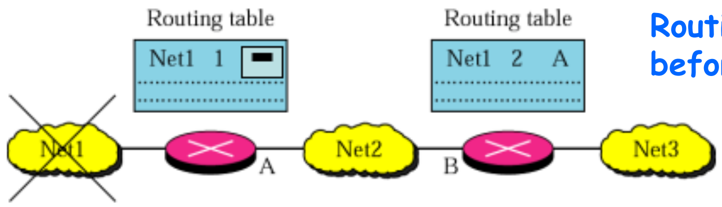
RIP - slow convergence



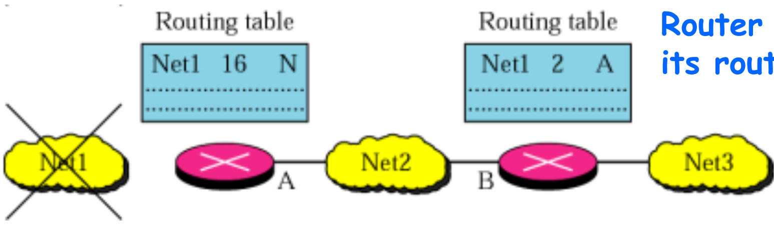
CSE, UNSW

57

RIP - instability problem (1)



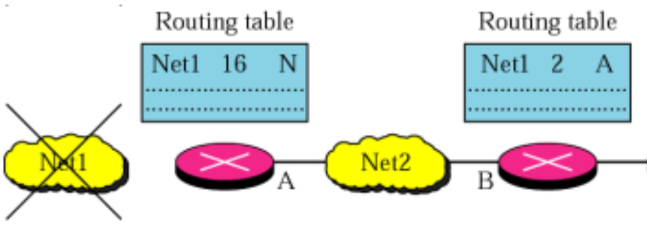
Routing tables before Net1 fails



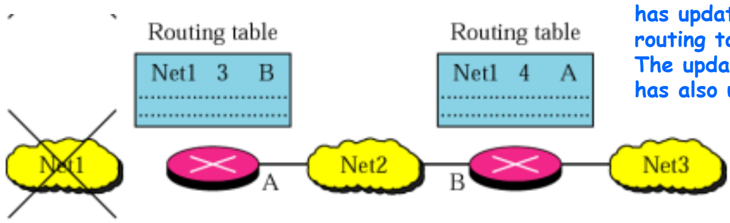
Router A updates its routing table

Exercise: If router B sends router A its routing table now, what will the routing table of A after the update?

Solution

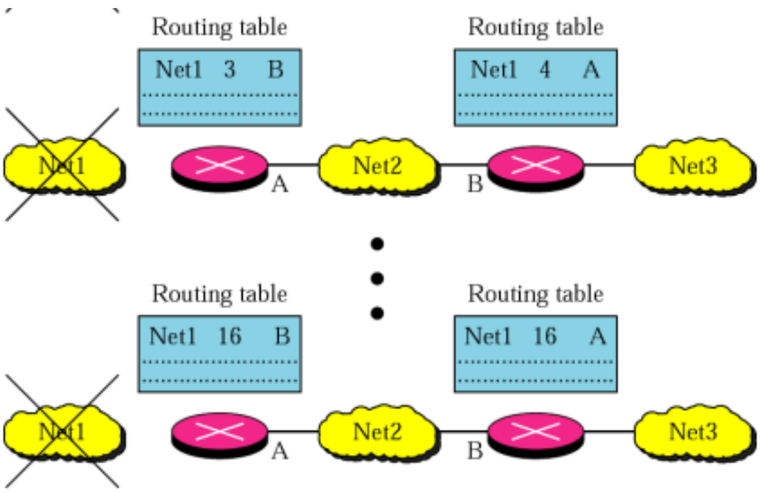


Before router B sends its routing table



After router A has updated its routing table, forwarded The update to B, and B has also updated its table

RIP - instability problem (2)



RIP instability problem (3)

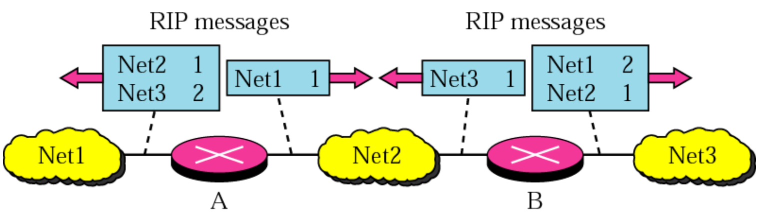
- Also known as count to infinity problem
- Remedy: Hop limit of 16
  - Note this limits the problem but doesn't solve it
  - This means RIP can only be used for small network
- Other remedies are based on modifying the protocol

RIP instability problem (3)

- Other remedies
  - Triggered update
    - *Almost immediate (a small random waiting recommended)* update, without waiting for periodic timer to go off, after change in routing table (Note: In the example before, the problem occurred because router B sends its update before router A)
    - Send only the entries that changed (not the whole table)
    - Regular update still occurs when periodic timer goes off
    - May not guarantee prevention of routing loop
    - Not enabled by default (has to be configured) - has traffic overhead implication
  - Split horizon [next slide]
  - Poison reverse [the slide after next]

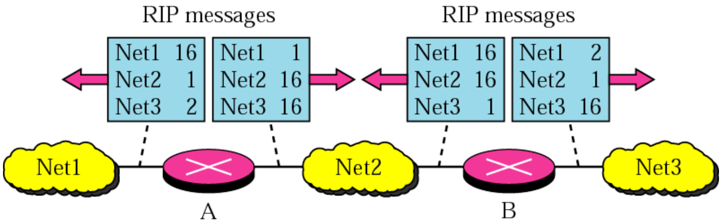
Split horizon

- Do not send update of a route through an interface over which the route information was originally received



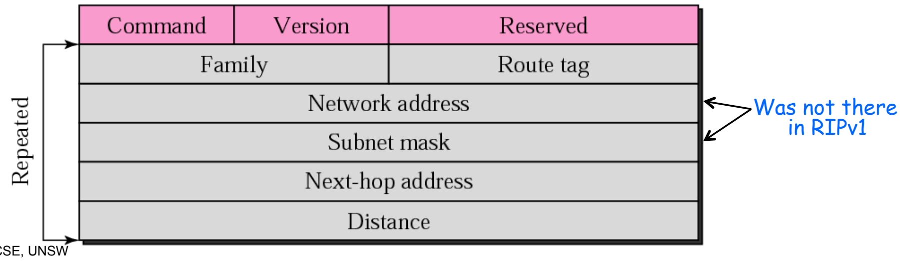
Poison reverse

- When sending update of a route through an interface over which the route information appeared originally, use hop count of 16



RIP version 2

- Subnet mask supports (subnetting can be used)
  - CIDR (as well as variable length subnets)
- Supports multicasting (RIP1 is broadcasted)
  - A host will not have to process RIP if it doesn't want to (e.g. not participating in RIP routing), reducing routing-related processing load
- Includes next-hop (may prevent unnecessary hops – how?)



RIPng for IPv6

- Designed to work with IPv6
  - E.g., no native authentication support (uses security features of IPv6)
  - RIPv2 supports native authentication, but RIPv1 has no authentication support
- No change in basic features and characteristics

RIP modes of operation

- Active mode (routers)
  - Both receive and advertise routing tables
- Passive mode (workstations)
  - Receive, but not advertise

Limitations of RIP

irrespective of the versions

- Network span (diameter) limited by maximum hop count of 15
  - Note that actual number of networks can be > 15!
  - *Network span* is not equivalent to *number of networks* in an autonomous system (hint:one router may connect to many networks, not just two)
  - RIP can only be applied to small ASs
- Cannot achieve a route alternative to shortest path (because *hop count* is used as metric)
- Slow convergence
- Large update message
  - Size of update message is proportional to the number of networks
- Too much traffic due to periodic update

References

- IBM Redbook (Section 4.3)
- Forouzan (Chapter 14 of 3rd Ed)