CS321: Computer Networks

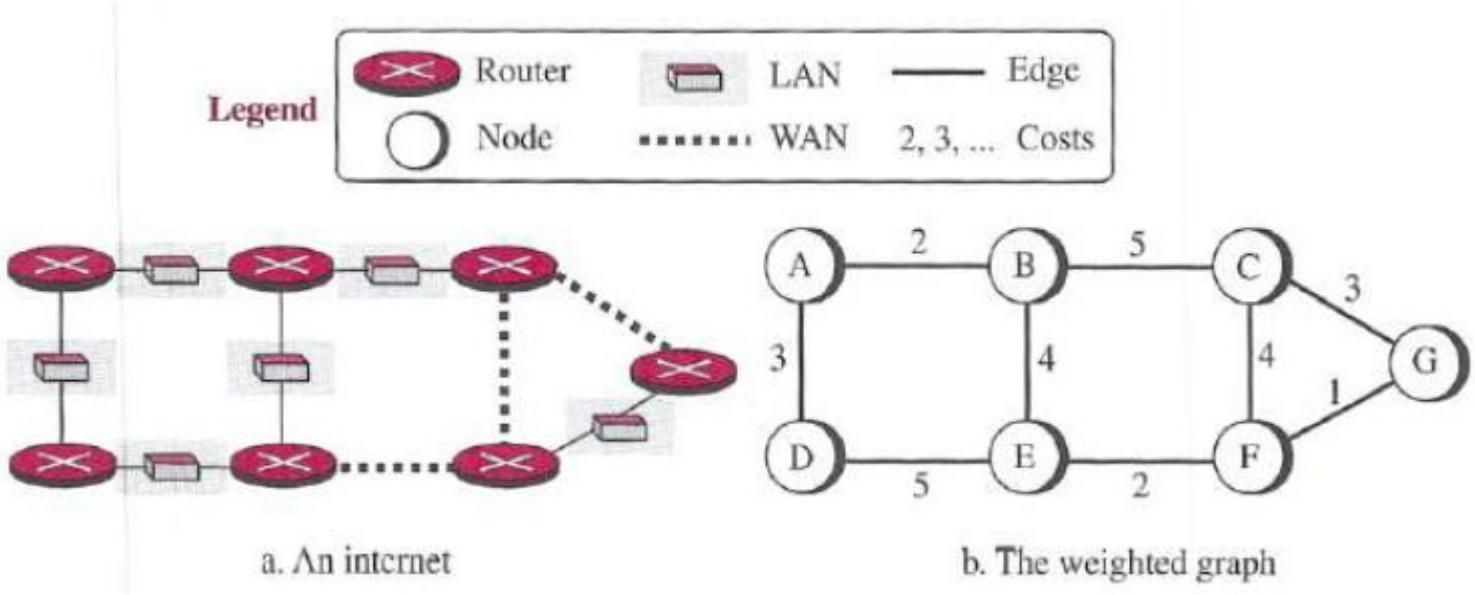
Unicast Routing

Dr. Manas Khatua
Assistant Professor
Dept. of CSE
IIT Jodhpur
E-mail: manaskhatua@iitj.ac.in

Introduction

- The goal of the network layer is deliver a datagram from its source to its destination.

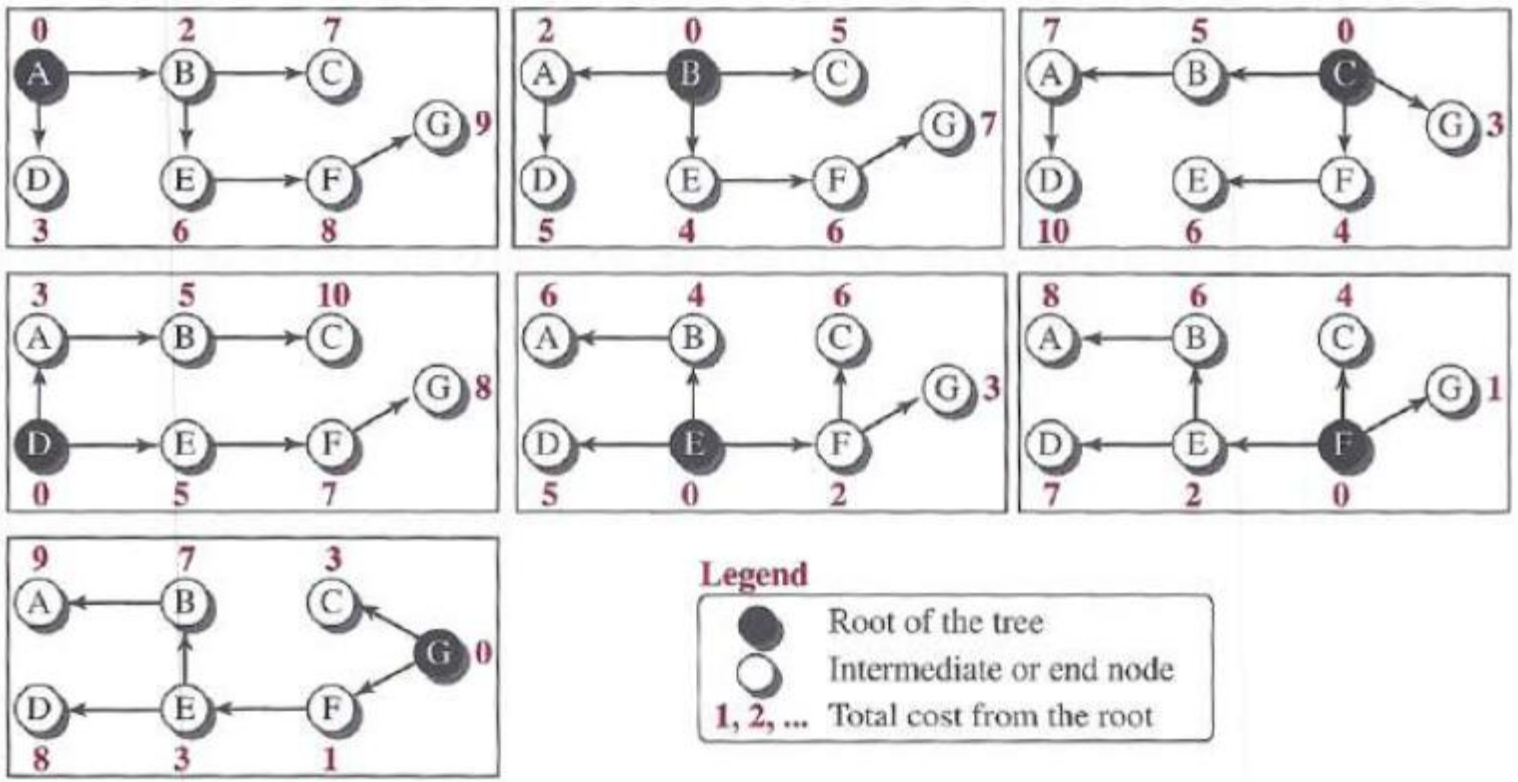
- Treat the Internet as a Graph



a. An internet

b. The weighted graph

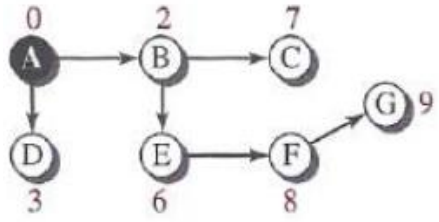
Least cost routing



- one of the ways to interpret the *best* route from the source router to the destination router is to find the *least cost* between the two.
  - Distance vector approach : Decentralized routing approach
  - Link state approach : Global routing approach

Distance Vector Routing

- a router continuously tells all of its neighbours what it knows about the whole internet (although the knowledge can be incomplete)
- each node creates its own least-cost tree with the (incomplete) information it receives from neighbours
- It is iterative, asynchronous, and distributed
- The heart of DVR is Bellman-Ford equation: $d_x(y) = \min_v\{c(x,v) + d_v(y)\}$
- Called dynamic routing algorithm.
- It contains one entry for each router in the subnet.



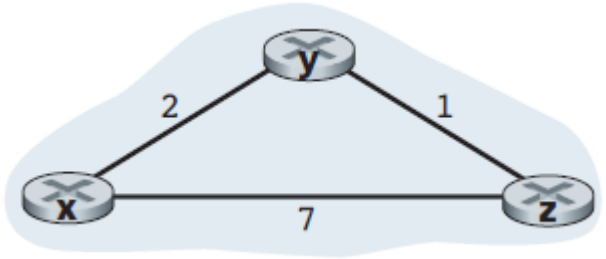
a. Tree for node A

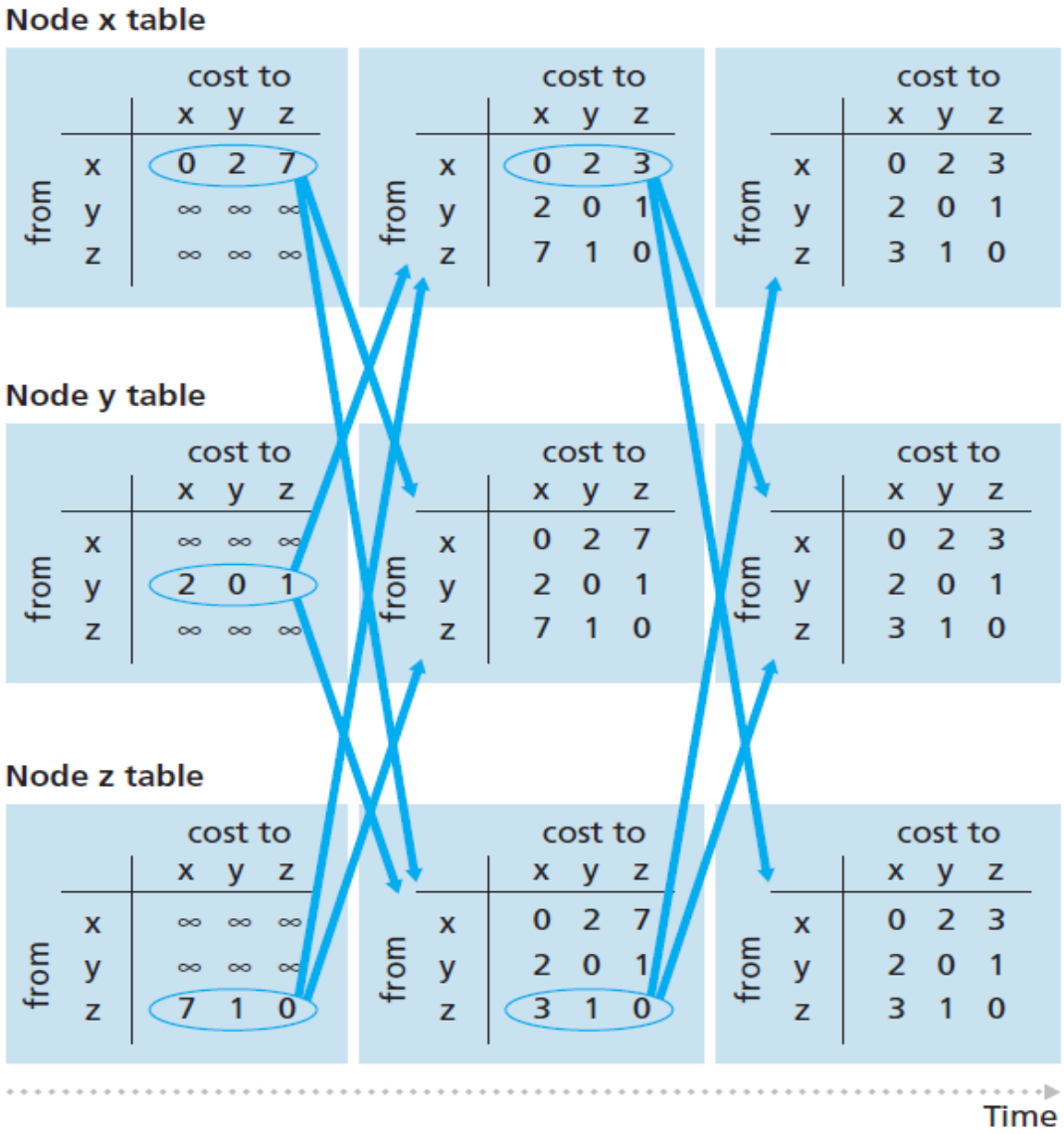
	A
A	0
B	2
C	7
D	3
E	6
F	8
G	9

b. Distance vector for node A

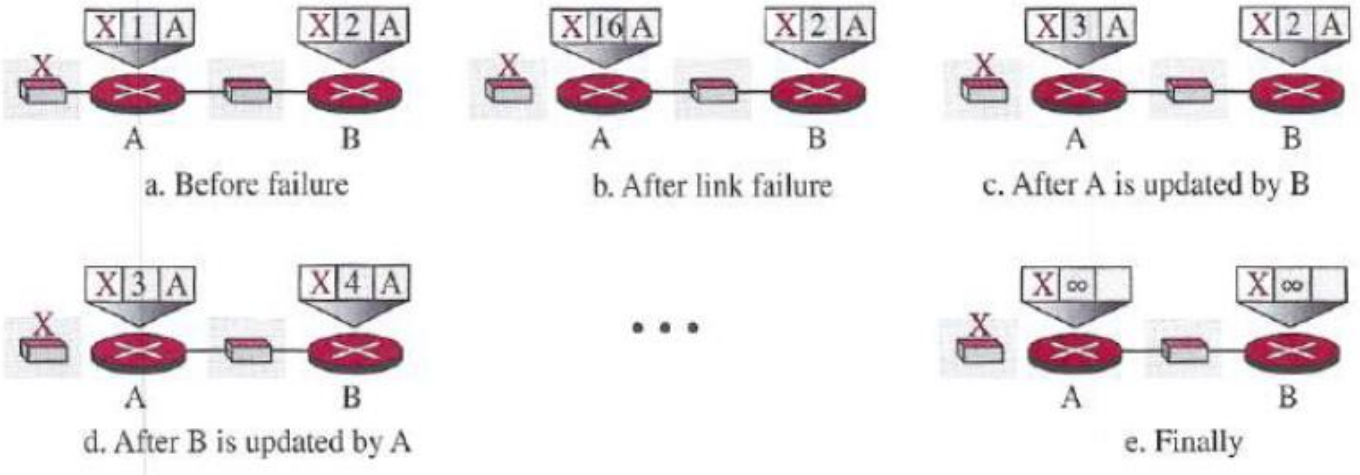
distance vector:
a one-dimensional array to represent the least-cost tree

Cont…


y
2
1
x
7
z


Node x table
cost to
x y z
from
x 0 2 7
y ∞ ∞ ∞
z ∞ ∞ ∞
x 0 2 3
y 2 0 1
z 7 1 0
x 0 2 3
y 2 0 1
z 3 1 0
Node y table
x ∞ ∞ ∞
y 2 0 1
z ∞ ∞ ∞
x 0 2 7
y 2 0 1
z 7 1 0
x 0 2 3
y 2 0 1
z 3 1 0
Node z table
x ∞ ∞ ∞
y ∞ ∞ ∞
z 7 1 0
x 0 2 7
y 2 0 1
z 3 1 0
x 0 2 3
y 2 0 1
z 3 1 0
Time

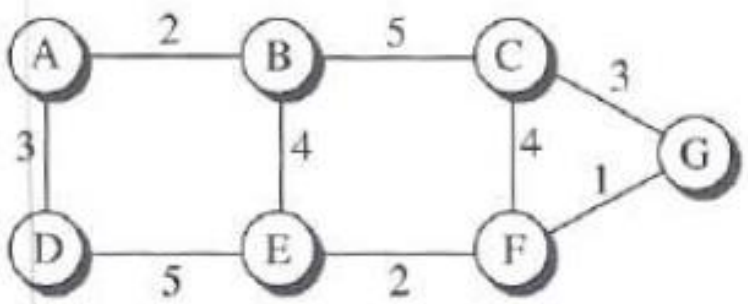
Count to Infinity Problem



- Solutions:
  - Split Horizon: each node sends only specific part of its table through each interface. For routers to send information only to the neighbors that are not exclusive links to the destination.
    - Route deleted problem due to timer
  - Poison Reverse: “Do not use this value; what I know about this route comes from you”

Link State Routing

- a router continuously tells all nodes what it knows about the neighbours

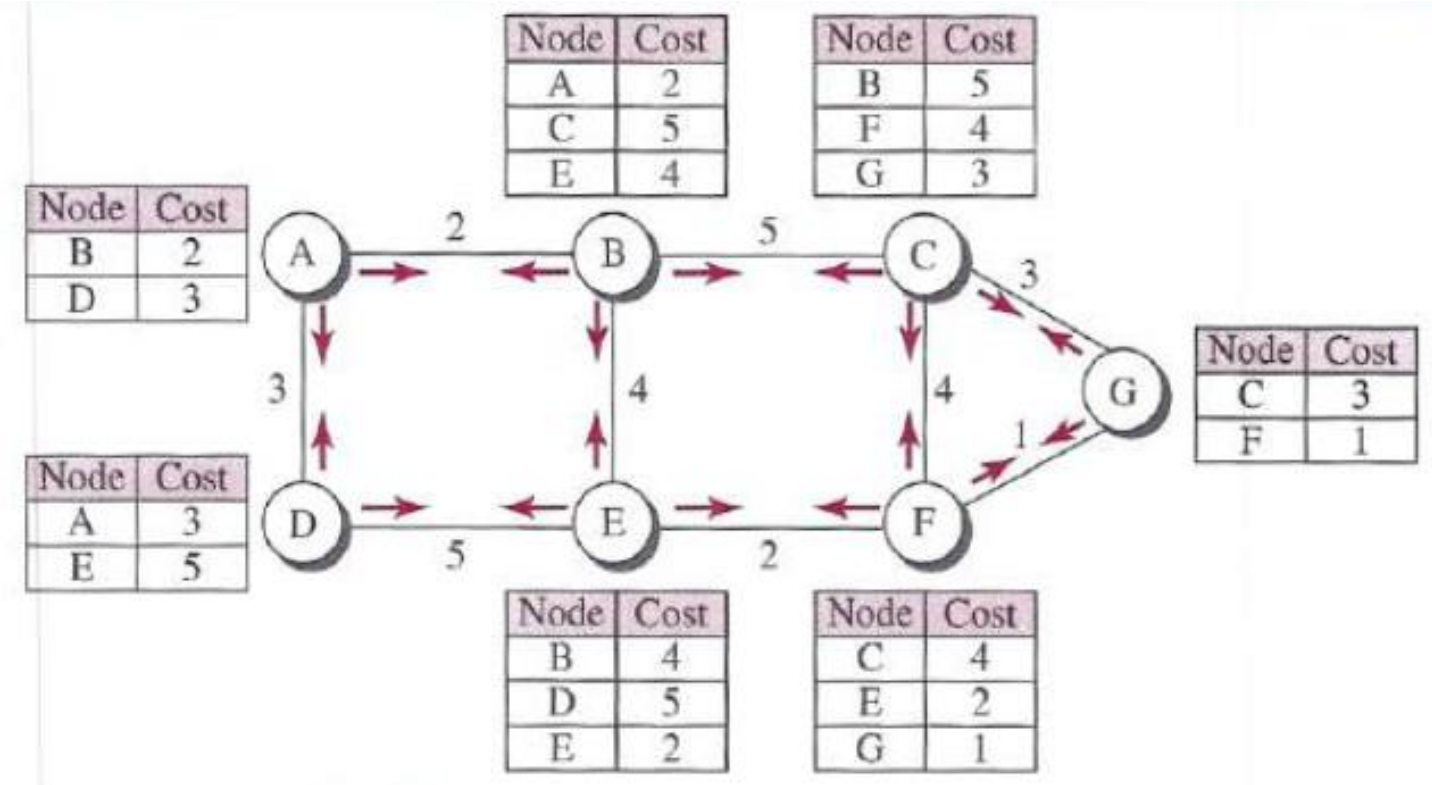


a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

Cont…



- To create a least-cost tree for itself
  - each node needs to run the famous Dijkstra Algorithm for computing single source least cost path.

DV v/s LS Routing

- *Message complexity:*
  - LS requires each node to know the cost of each link in the network.
  - This requires O(|N| |E|) messages to be sent.
  - Also, whenever a link cost changes, the new link cost must be sent to all nodes.
  - The DV algorithm requires message exchanges between directly connected neighbors at each iteration.

- *Speed of convergence:*
  - Implementation of LS is an O(|N|2) algorithm requiring O(|N| |E|)) messages.
  - The DV algorithm can converge slowly and can have routing loops while the algorithm is converging.
  - The DV also suffers from the count-to-infinity problem.

- *Robustness:*
  - an LS node is computing only its own forwarding tables
  - This means route calculations are somewhat separated under LS, providing a degree of robustness.
  - Under DV, a node can advertise incorrect least-cost paths to any or all destinations.
  - In this sense, an incorrect node calculation can be diffused through the entire network under DV.

Path-Vector Routing

- LS and DV both are based on least-cost routing
  - Does not allow the sender to apply its own policy

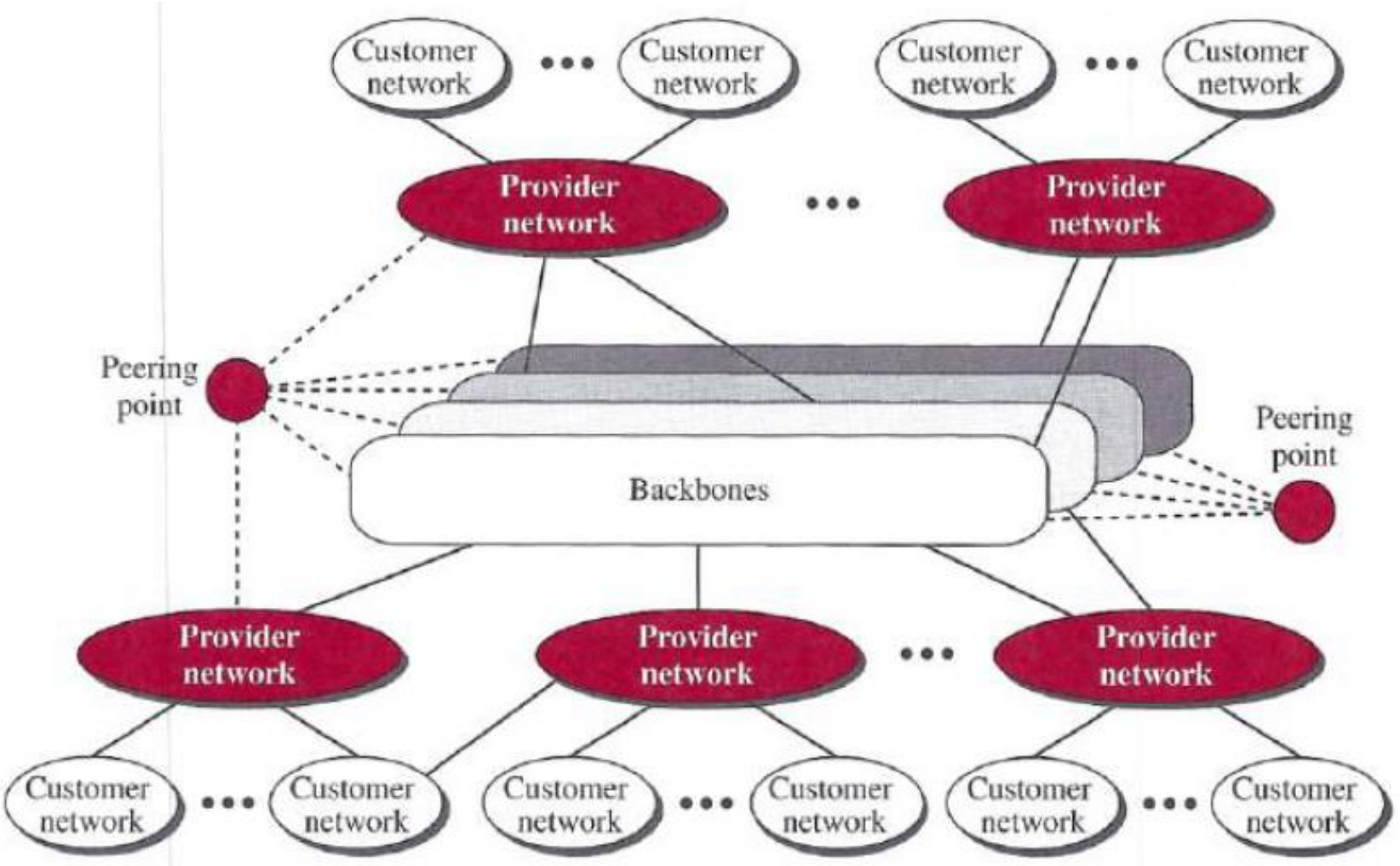
- PV based on best path according to desired policy
  - mainly used in routing between ISPs

- Path is determined from source to destination using spanning tree (might not be least cost)

- Spanning trees are made gradually and asynchronously likewise DVR

- Follow Bellman-Ford, but not the least-cost concept

Internet Structure



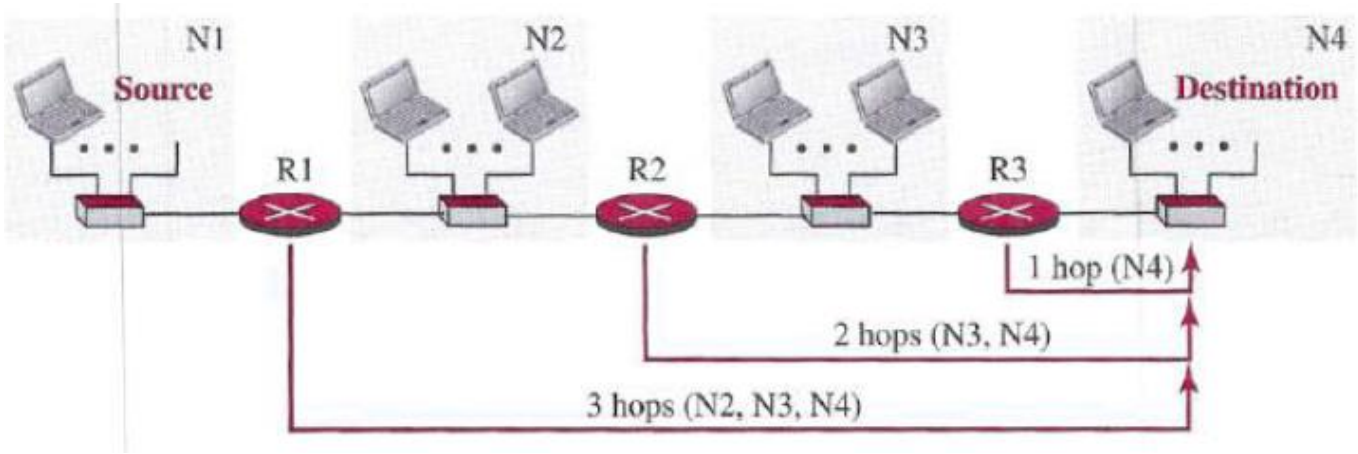
Routing Protocols

- Each ISP is considered as Autonomous System (AS)
  - Stub AS: connects to one other AS only
  - Multihomed AS: connects more than one AS, but refuses to carry transit traffic
  - Transit AS: connects more than one other AS, and carries local and transit traffic

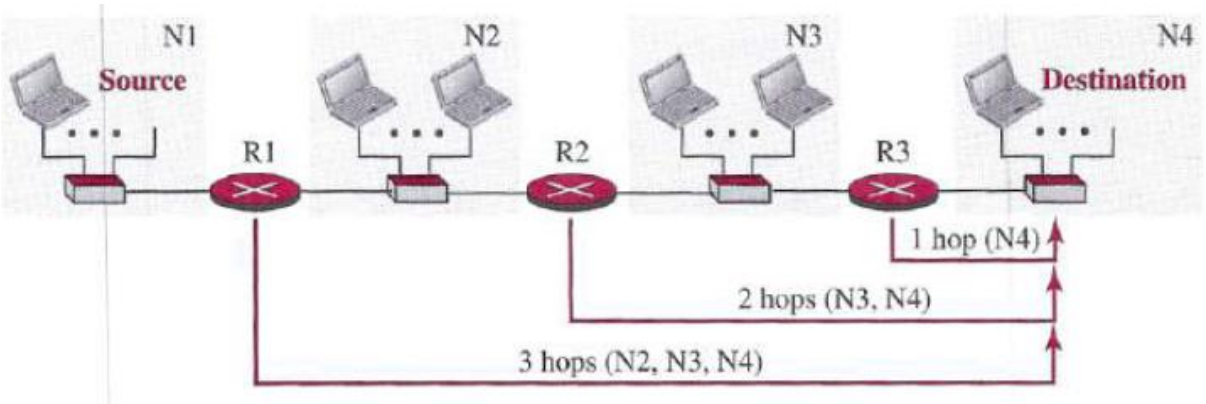
- Two types of routing protocol:
  - Intra-AS / intradomain / Interior Gateway Protocol (IGP)
    - E.g., RIP (Routing Information Protocol) - use DVR
    - OSPF (Open Shortest Path First) - use LSR
  - Inter-AS / interdomain / Exterior Gateway Protocol (EGP)
    - E.g., BGP (Border Gateway Protocol) - use PVR

RIP

- Follow distance-vector routing with the following modifications
  - Router advertise the cost to reach different networks instead of individual node
  - Cost is defined as number of hops
  - Advertise forwarding table instead of distance-vector



Cont…



- Final Forwarding Tables

Forwarding table for R1

Destination network	Next router	Cost in hops
N1	——	1
N2	——	1
N3	R2	2
N4	R2	3

Forwarding table for R2

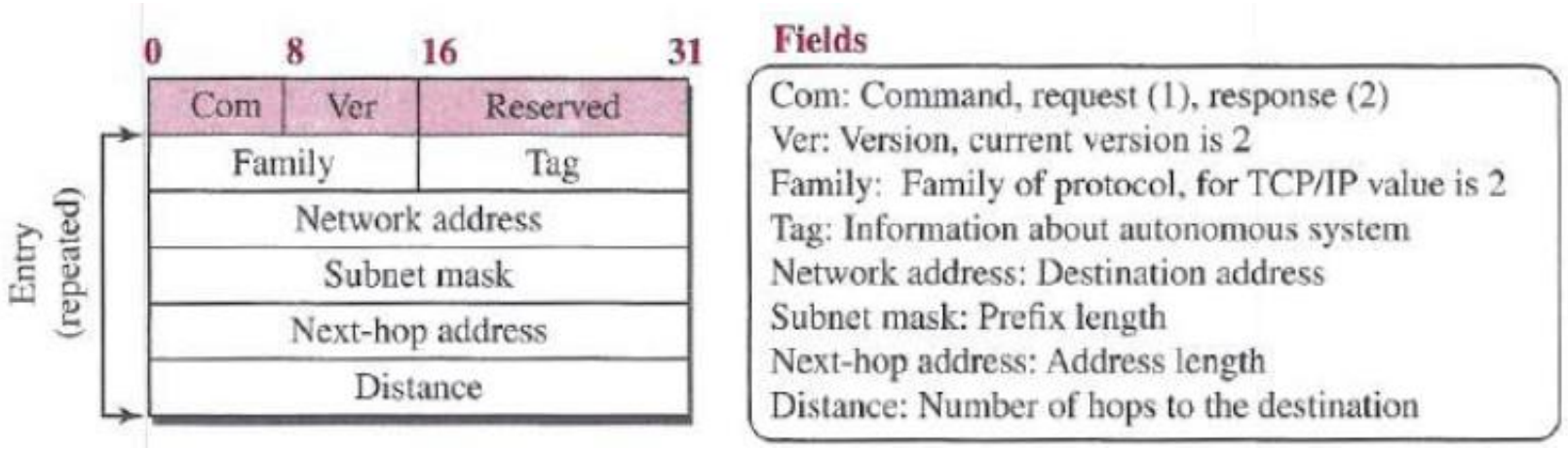
Destination network	Next router	Cost in hops
N1	R1	2
N2	——	1
N3	——	1
N4	R3	2

Forwarding table for R3

Destination network	Next router	Cost in hops
N1	R2	3
N2	R2	2
N3	——	1
N4	——	1

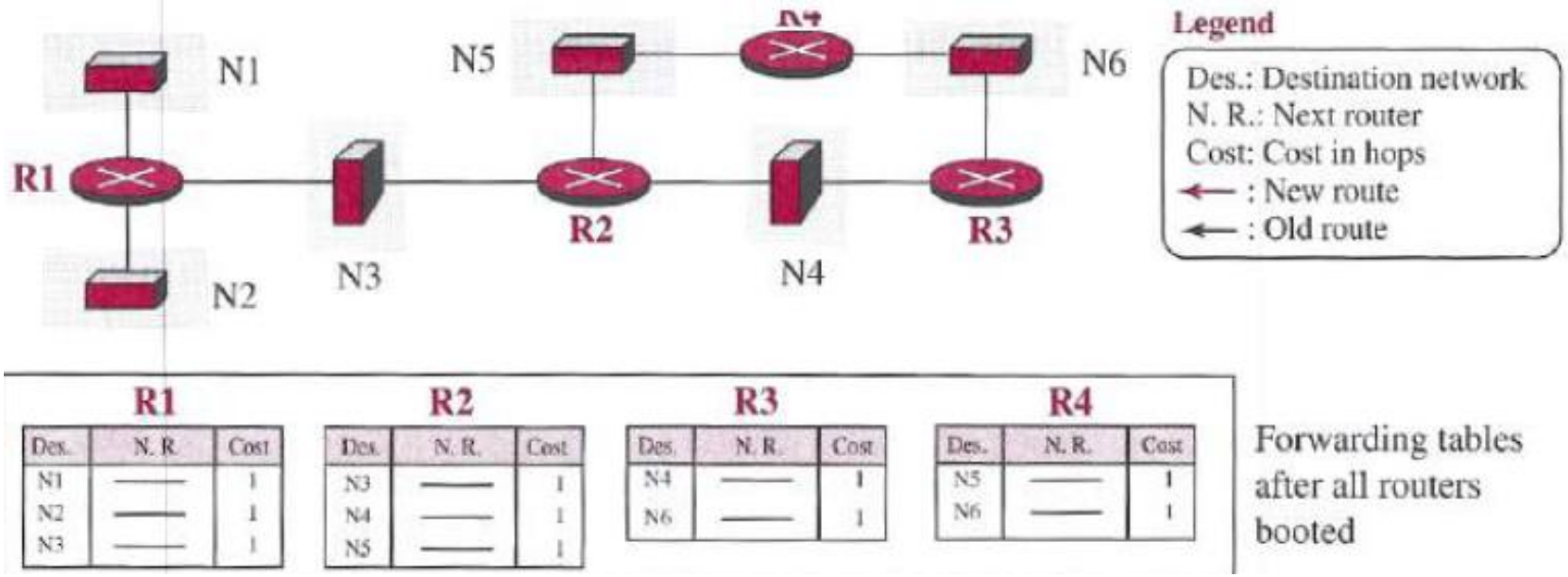
RIP Implementation

- RIP runs at the application layer but creates forwarding table for network layer
- Uses the service of UDP on port 520
- Runs as background process
  - Two processes : a client and a server
- RIP uses two types of messages: request and response



RIP Algorithm

- RIP uses timers:
  - Update timer: for advertising update message regularly. default value 30 sec.
  - Expiration timer / Invalid timer : specifies how long a routing entry can be in the routing table without being updated. Default value 180 sec.
  - Flush Timer: The flush timer controls the time between the route is invalidated or marked as unreachable and removal of entry from the routing table. Default time 240 sec.
  - Hold-down Timer: This allows the route to get stabilized. During this time no update can be done to that routing entry. Default value 180 sec.



R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1

R2

Des.	N. R.	Cost
N3	—	1
N4	—	1
N5	—	1

R3

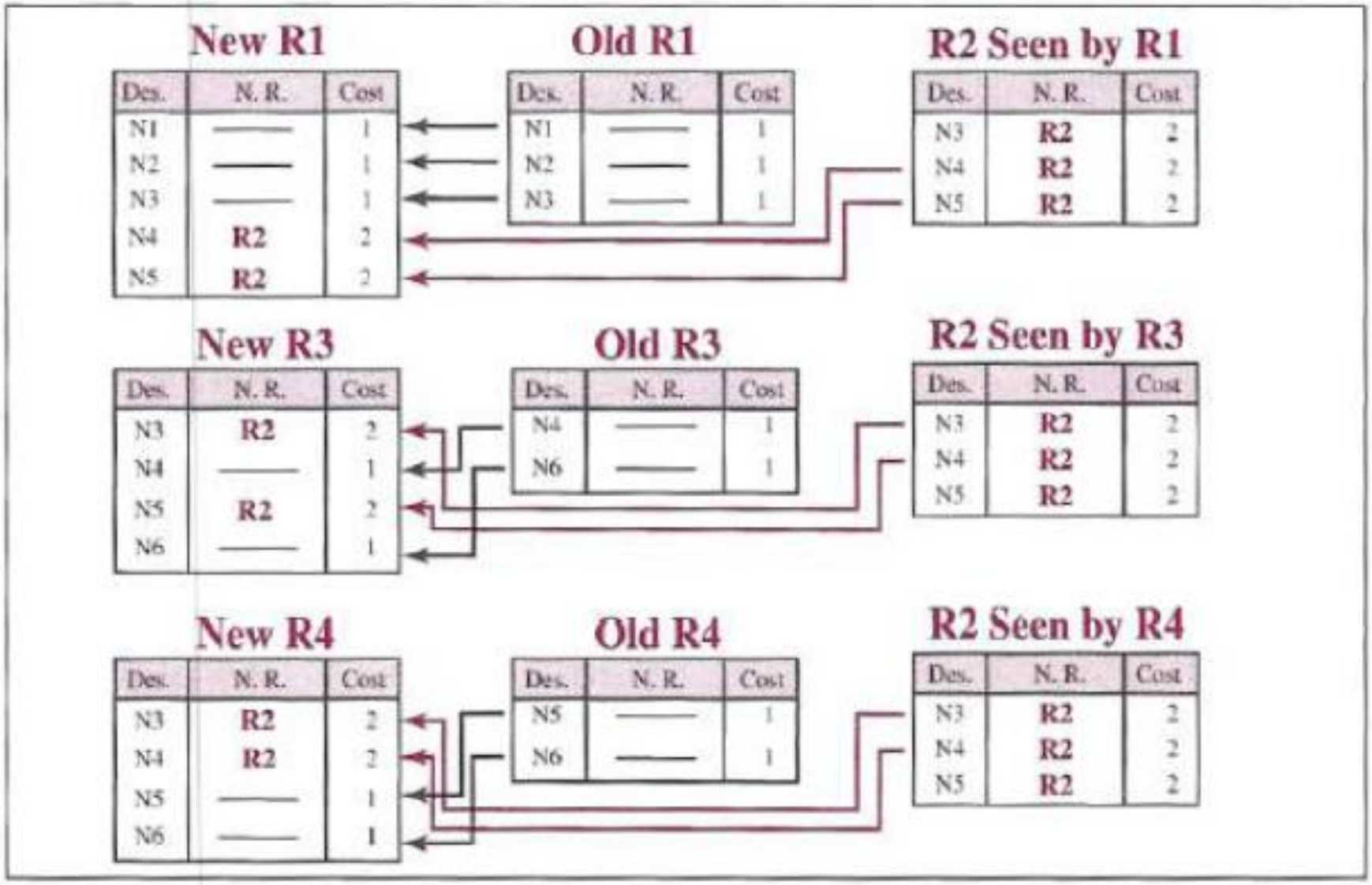
Des.	N. R.	Cost
N4	—	1
N6	—	1

R4

Des.	N. R.	Cost
N5	—	1
N6	—	1

Forwarding tables after all routers booted

Cont…


New R1
Des. N. R. Cost
N1 — 1
N2 — 1
N3 — 1
N4 R2 2
N5 R2 2
Old R1
Des. N. R. Cost
N1 — 1
N2 — 1
N3 — 1
R2 Seen by R1
Des. N. R. Cost
N3 R2 2
N4 R2 2
N5 R2 2
New R3
Des. N. R. Cost
N3 R2 2
N4 — 1
N5 R2 2
N6 — 1
Old R3
Des. N. R. Cost
N4 — 1
N6 — 1
R2 Seen by R3
Des. N. R. Cost
N3 R2 2
N4 R2 2
N5 R2 2
New R4
Des. N. R. Cost
N3 R2 2
N4 R2 2
N5 — 1
N6 — 1
Old R4
Des. N. R. Cost
N5 — 1
N6 — 1
R2 Seen by R4
Des. N. R. Cost
N3 R2 2
N4 R2 2
N5 R2 2

OSPF

Each OSPF router can create a forwarding table after finding the shortest-path tree between itself and the destination using Dijkstra's algorithm

Figure 20.19 *Metric in OSPF*

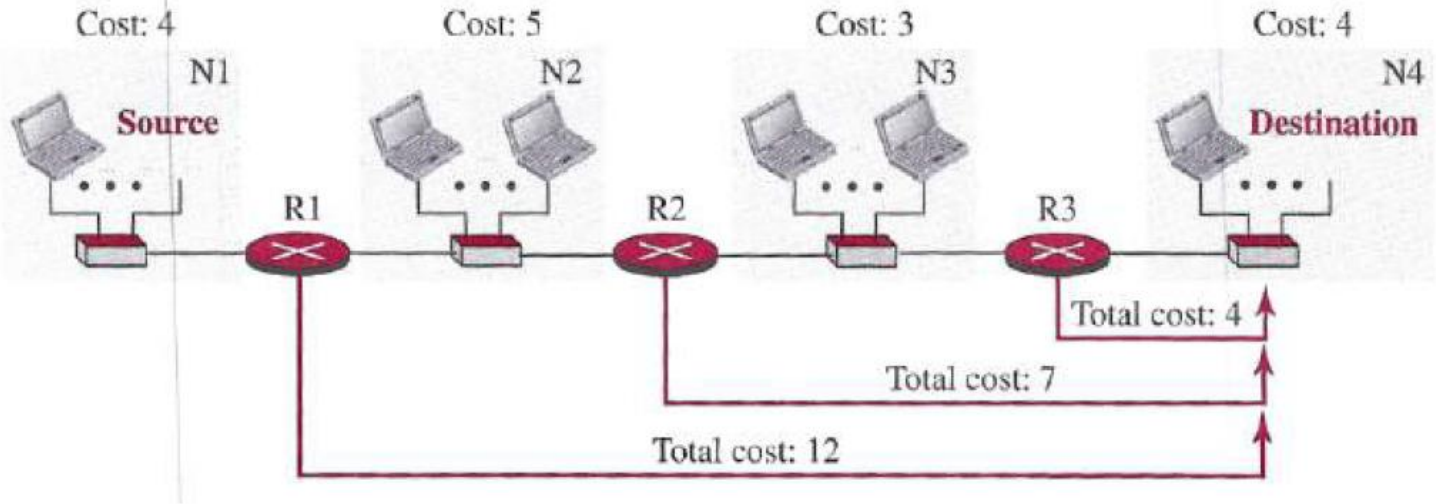


Figure 20.20 *Forwarding tables in OSPF*

Forwarding table for R1

Destination network	Next router	Cost
N1	—	4
N2	—	5
N3	R2	8
N4	R2	12

Forwarding table for R2

Destination network	Next router	Cost
N1	R1	9
N2	—	5
N3	—	3
N4	R3	7

Forwarding table for R3

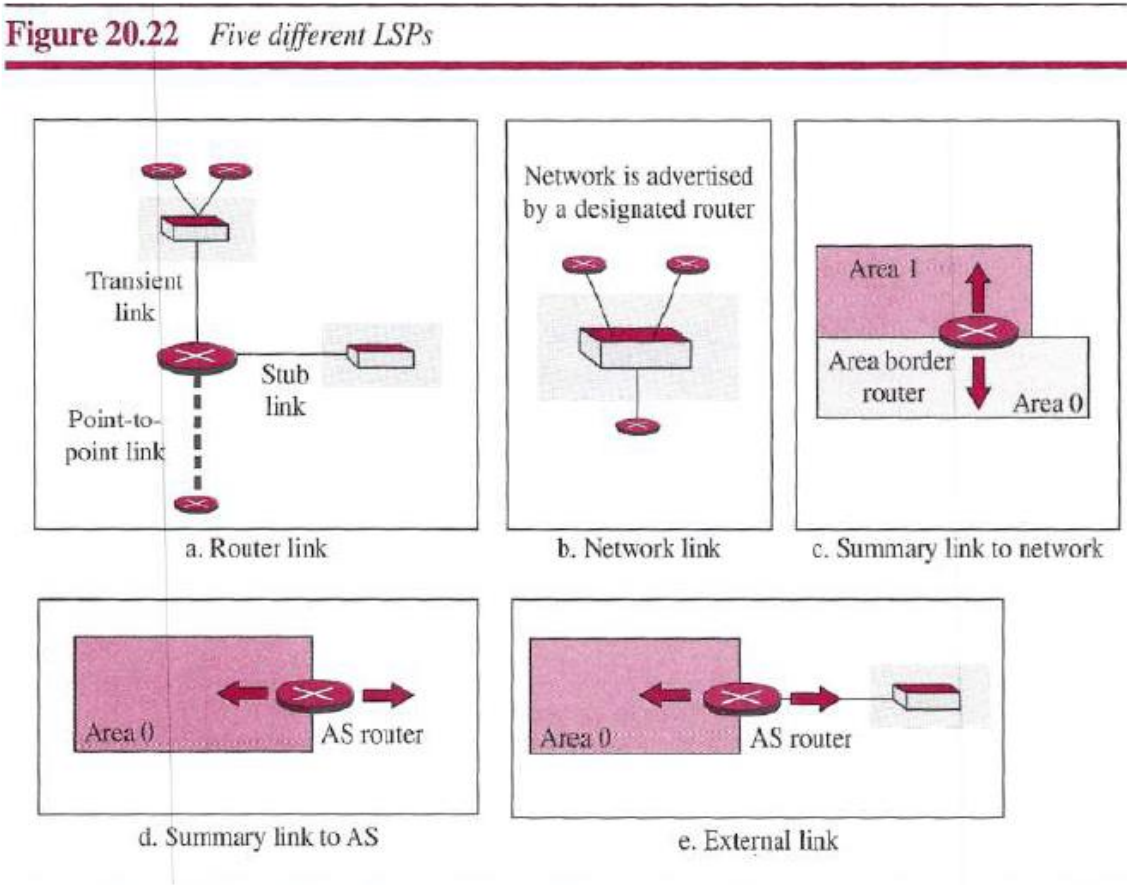
Destination network	Next router	Cost
N1	R2	12
N2	R2	8
N3	—	3
N4	—	4

Link-State Advertisement

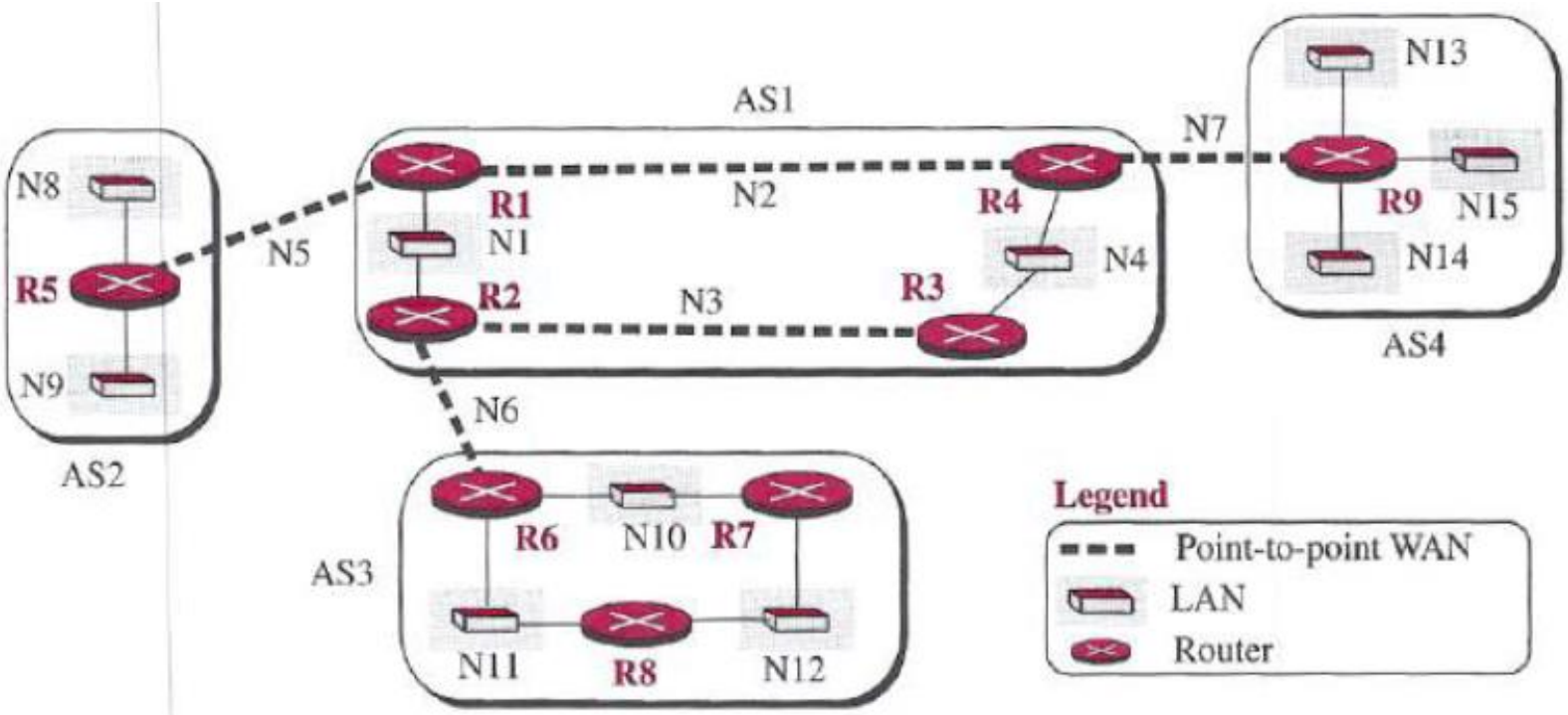
- a router advertise the state of each link to all neighbors for the formation of the LSDB.

- We can have five types link-state advertisements:
  - router link,
  - network link,
  - summary link to network,
  - summary link to AS:
    border router,
  - external link.

- OSPF messages
  - *hello* message
  - *database description* message
  - *link-state request* message
  - *link-state update* message
  - *link-state ACK* message

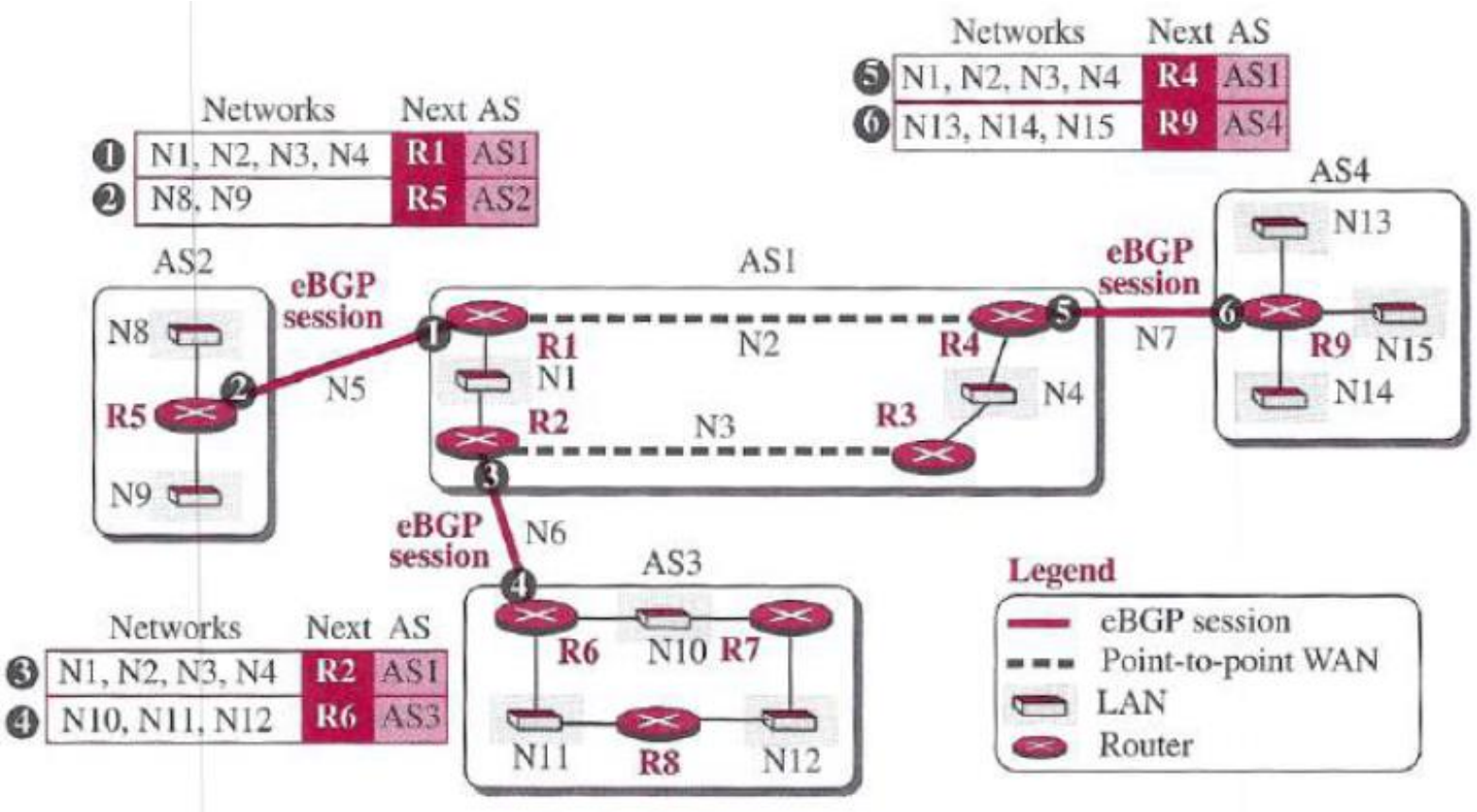


BGP



- AS2, AS3, AS4 : stub AS; AS1 : transient AS
- Each AS use RIP / OSPF for intra-domain routing
- BGP for inter-domain routing used by all AS
  - eBGP: on each border router
  - iBGP: on all routers

External BGP (eBGP)



- 3 pairs: R1-R5, R4-R9, R2-R6 => 3 eBGP sessions (using TCP)
- Message 1 is sent by router R1 and tells router R5 that N1, N2, N3, and N4 can be reached through router R1. Then, R5 updates its table.

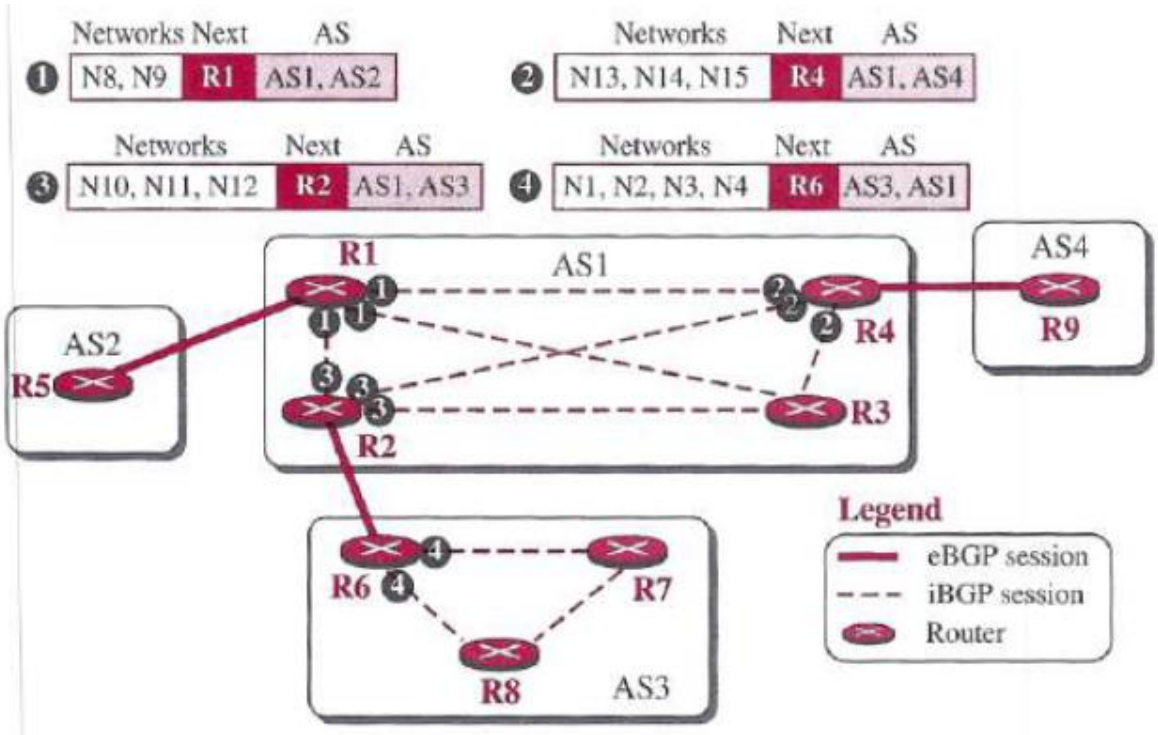
Limitation in eBGP

1. Border router does not know how to route a packet destined for non-neighbour AS.
   - E.g., R5 does not know about networks in AS3, AS4

2. None of the non-border routers know how to route a packet destined for any network in other ASs.
   - E.g., R3 does not know about networks in AS2, AS3, AS4

Solution: Internal BGP (iBGP)

Internal BGP (iBGP)

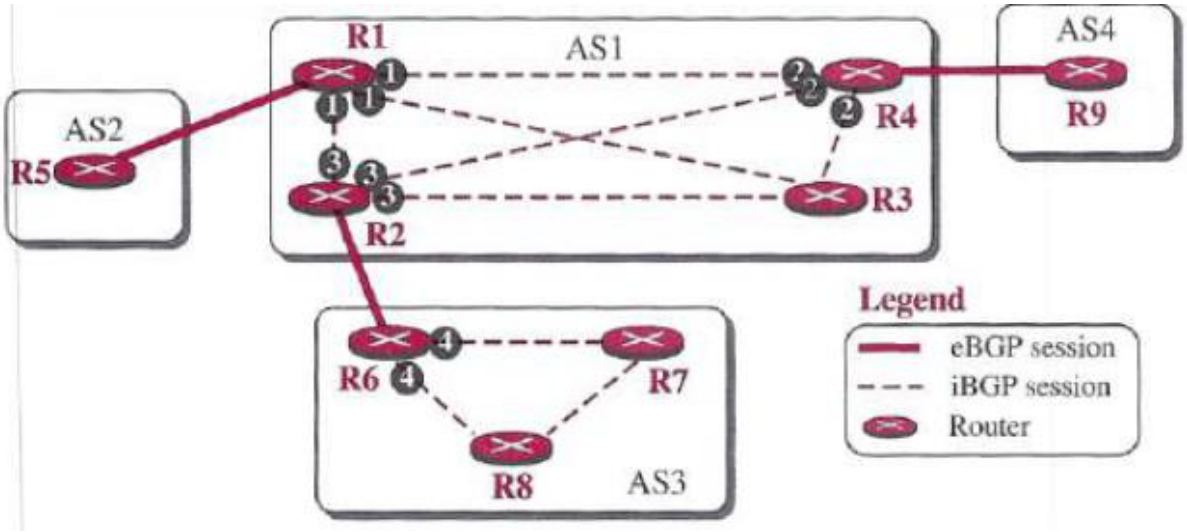


No message for R3, R7, R8

- For n router in an AS: n(n-1)/2 sessions
- No iBGP session for single router in an AS.

- Message 1 sent by R1 tells that N8 and N9 are reachable through the path AS1-AS2, but the next router is R1.

Finalizing BGP path table



- R1 receives:

Networks	Next AS
N8, N9	R5 AS2

Networks	Next	AS
N10, N11, N12	R2	AS1, AS3

Networks	Next	AS
N13, N14, N15	R4	AS1, AS4

Networks	Next	Path
N8, N9	R5	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R1

Cont…

Networks	Next	Path
N8, N9	R5	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R1

Networks	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R6	AS1, AS3
N13, N14, N15	R1	AS1, AS4

Path table for R2

Networks	Next	Path
N8, N9	R2	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R3

Networks	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R1	AS1, AS3
N13, N14, N15	R9	AS1, AS4

Path table for R4

Networks	Next	Path
N1, N2, N3, N4	R1	AS2, AS1
N10, N11, N12	R1	AS2, AS1, AS3
N13, N14, N15	R1	AS2, AS1, AS4

Path table for R5

Networks	Next	Path
N1, N2, N3, N4	R2	AS3, AS1
N8, N9	R2	AS3, AS1, AS2
N13, N14, N15	R2	AS3, AS1, AS4

Path table for R6

Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R7

Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R8

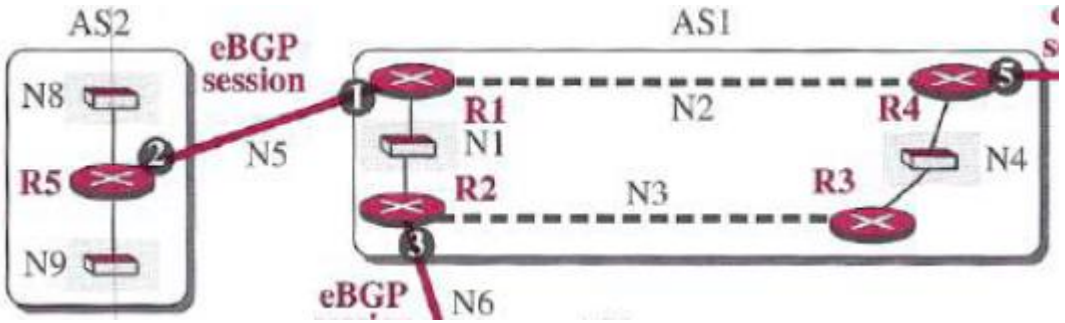
Networks	Next	Path
N1, N2, N3, N4	R4	AS4, AS1
N8, N9	R4	AS4, AS1, AS2
N10, N11, N12	R4	AS4, AS1, AS3

Path table for R9

Injection of Information into Forwarding Table

- The role of BGP is to help the routers inside the AS to augment their routing table
- Cost update problem for mixing of RIP and OSPF

- For stub AS:

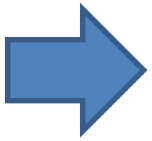


Des.	Next	Cost
N8	—	1
N9	—	1

Table for R5

Networks	Next	Path
N1, N2, N3, N4	R1	AS2, AS1
N10, N11, N12	R1	AS2, AS1, AS3
N13, N14, N15	R1	AS2, AS1, AS4

Path table for R5

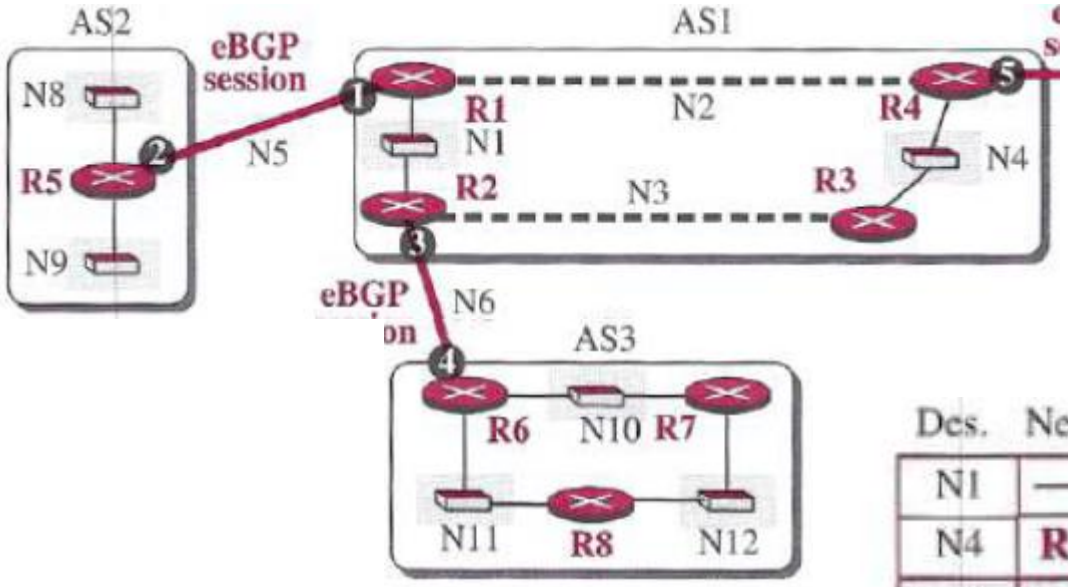


Des.	Next	Cost
N8	—	1
N9	—	1
0	R1	1

Table for R5

Cont…

For transient AS:



Des.	Next	Cost
N1	—	1
N4	R4	2

Table for R1

Networks	Next	Path
N8, N9	R5	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R1

Des.	Next	Cost
N1	—	1
N4	R4	2
N8	R5	1
N9	R5	1
N10	R2	2
N11	R2	2
N12	R2	2
N13	R4	2
N14	R4	2
N15	R4	2

Table for R1

BGP Messages

- BGP uses 4 types of messages
  - Open Message : for creating neighbourhood relationship. Router creates a TCP connection with neighbour router and sends Open message.
  - Update Message: to withdraw destinations that have been advertised previously or to announce a route to a new destination
  - Keepalive Message: BGP peers exchange this message to tell each other that they are alive
  - Notification: when an error is detected or the router wants to close the session

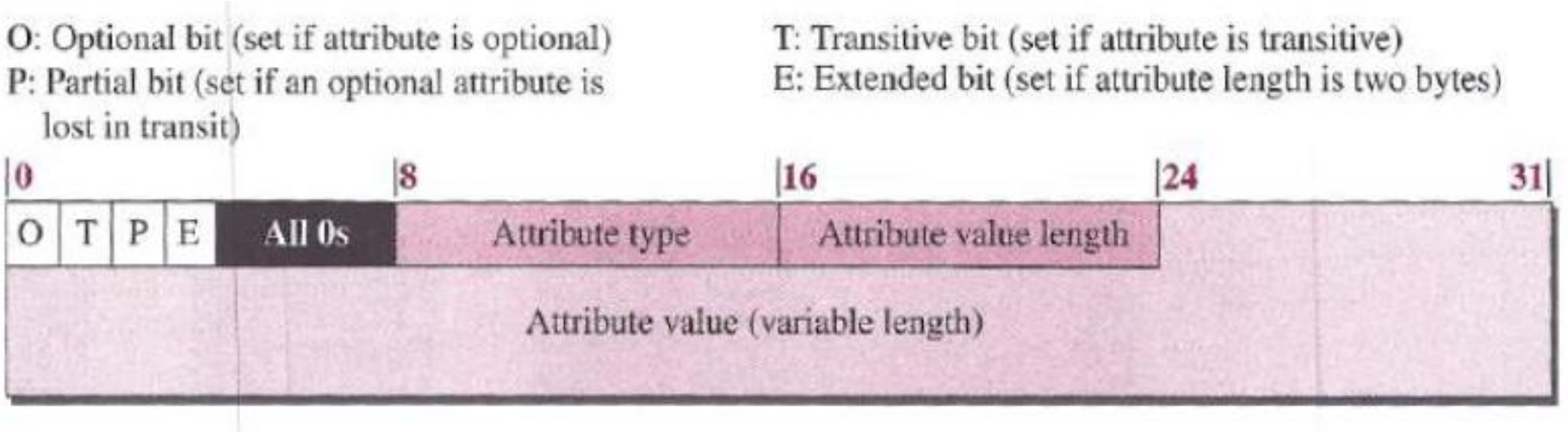
BGP Path Attributes

- In RIP/OSPF, a destination has two associated information:
  - next hop and
  - cost

- But, inter-domain routing needs more information

- In BGP, those information are called path attributes
  - Destination can be associated with 7 attributes
  - Two types of attribute: mandatory and optional
  - Mandatory attributes must be present in each UPDATE message
  - Optional attributes type: transitive and nontransitive

Cont…



- ORIGIN (Type 1): This is mandatory attribute.
  - Value=1: path information comes from RIP/OSPF
  - Value=2: information comes from BGP
  - Value=3: information comes from other sources

- MULTI-EXIT-DISC (Type 4): This is optional intransitive attribute
  - discriminates among multiple exit paths to a destination
  - This is intransitive as the best exit path (e.g. shortest path) is different from one AS to other

Thanks!