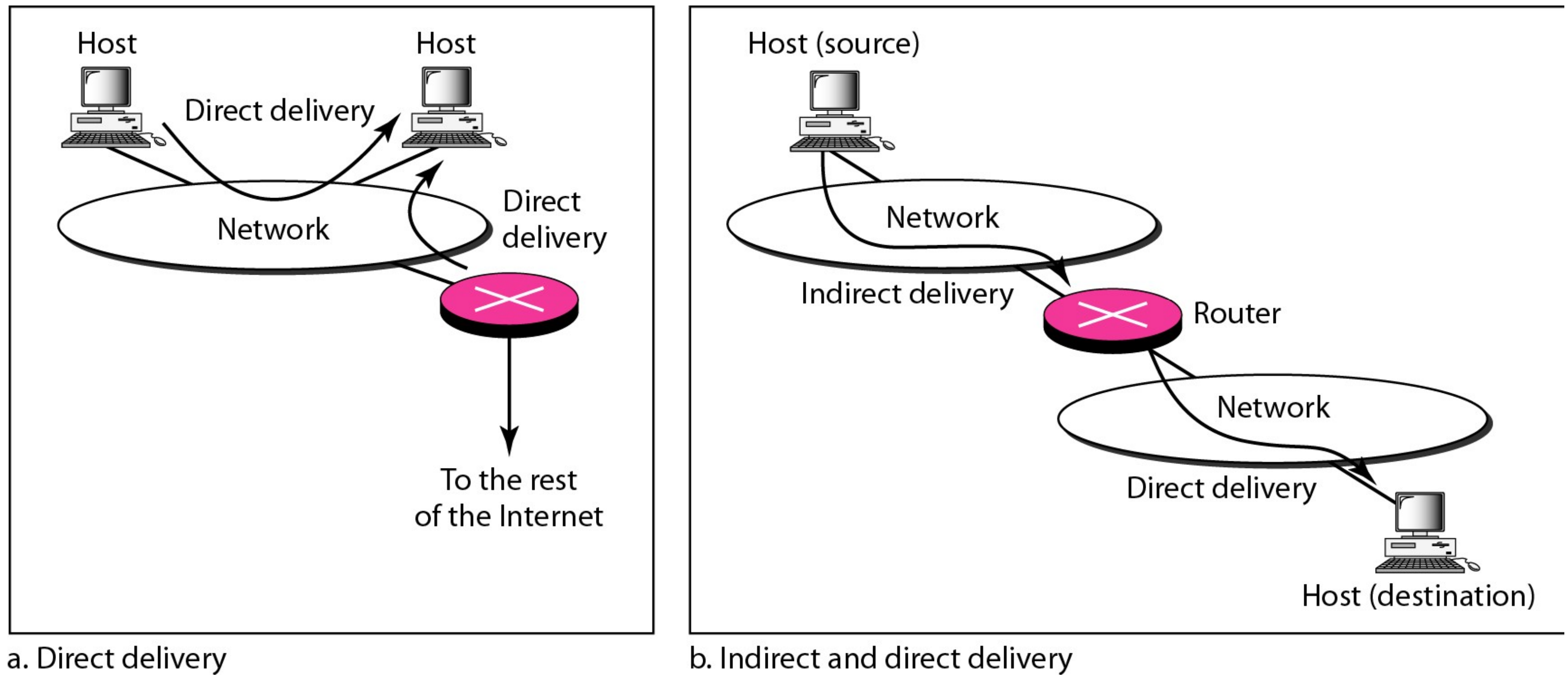
Data Communication and Computer Network

Routing

Routing

- The process of sending a message from a source to a destination through intermediate nodes
- A route – a path from the source to the destination
- Routing protocol / algorithm
  - Finds one or more routes between a source and a destination
  - Usually stores routes found in a routing table for future use, each node has a routing table
  - Routing table has to be updated if this node comes to know about new routes or changes in existing routes

Direct and indirect delivery



Goals of routing protocols

- Correctness
- Optimal
- Efficiency/Low overhead
- Robust (able to handle failure of node or link)
- Rapid convergence when network conditions change

- How to compare among routes?
  - Network usually modeled as a weighted graph
  - Cost of path in the weighted graph
  - Minimum number of hops (if all links have same cost)

When / where are routing decisions taken?

- When
  - During circuit setup (circuit switching / virtual circuit switching), or
  - During switching of each datagram (packet switching)

- Where
  - Distributed: made by each node when it receives a packet to forward, each node collects info from other nodes to learn best route
  - Centralized: one node decides routes for every node
  - Source routing: source node decides route and puts complete path to destination in packet

Route method versus next-hop method

Forwarding : Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

a. Routing tables based on route

b. Routing tables based on next hop

Routing table for host A

Destination	Route
Host B	R1, R2, host B

Destination	Next hop
Host B	R1

Routing table for R1

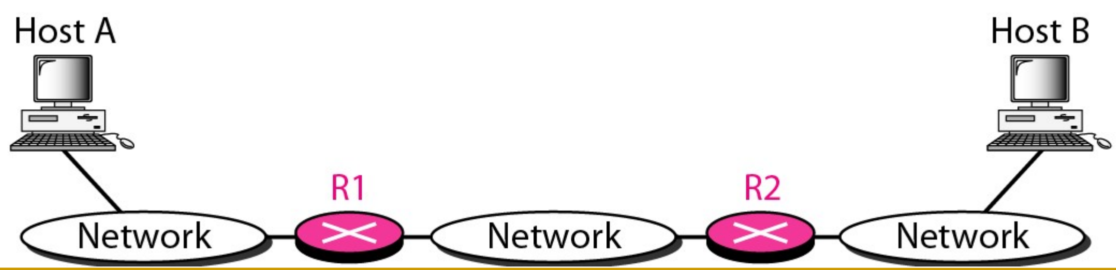
Destination	Route
Host B	R2, host B

Destination	Next hop
Host B	R2

Routing table for R2

Destination	Route
Host B	Host B

Destination	Next hop
Host B	---



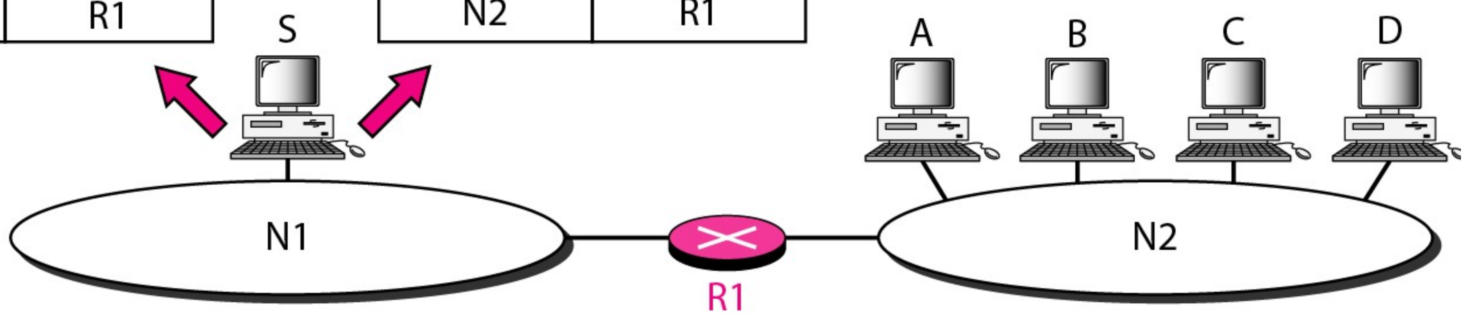
Host-specific versus network-specific method

Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific method

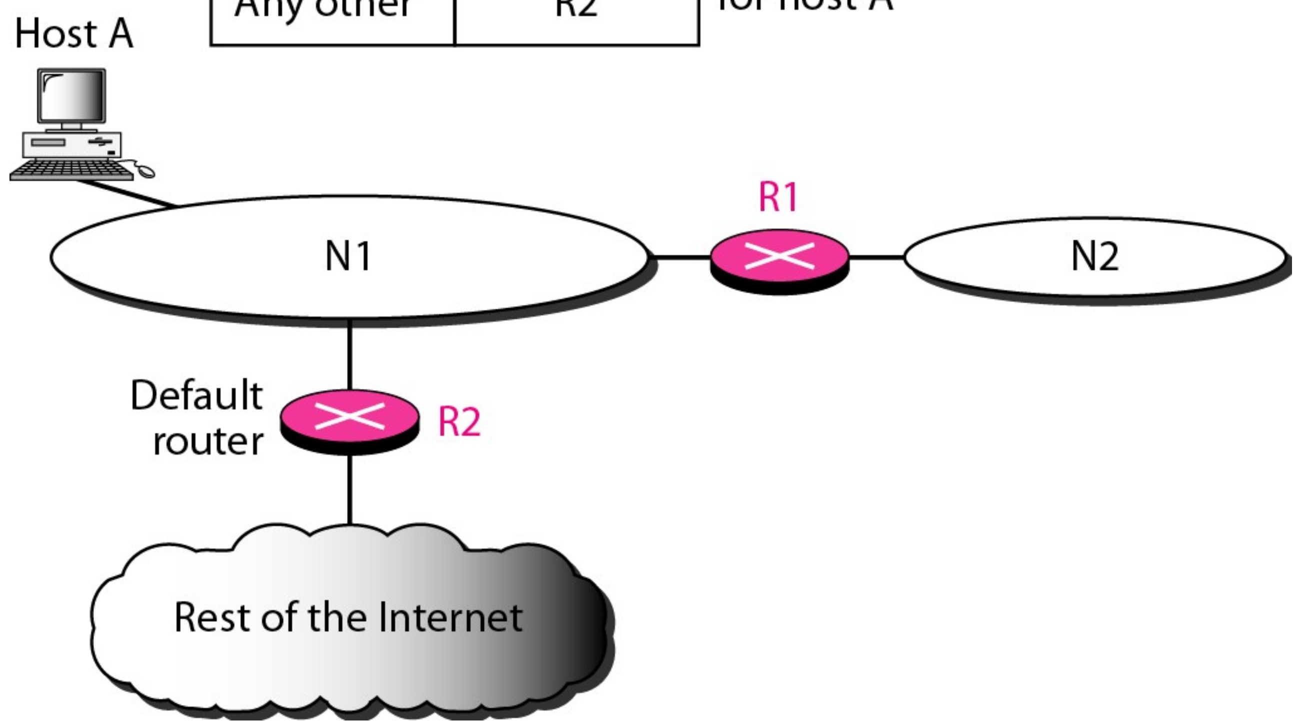
Destination	Next hop
N2	R1



Default method

Destination	Next hop
N2	R1
Any other	R2

Routing table for host A



Routing basics

- Routing table usually contains
  - Network addresses (in CIDR notation, or in explicit network and mask form)
  - Host-specific routes (32 bit prefixes) i.e. IP address of individual destination hosts
  - Default entry (both network and subnet mask 0.0.0.0)
  - And a next hop address for each of them

- Usually a routing cache used
  - Contains recent routing decisions
  - Destination address first looked up in cache
  - If not found, longest-prefix match done in routing table

Routing procedure (in CIDR)

Extract destination IP address D from the datagram

for each entry in routing table **do**

  N ← D bitwise AND subnet mask

  verify if N matches the network address of the entry

  # maybe partial match i.e. only some prefix bits match

end for

if no match found

  declare routing error

else if one or more matches found

  select match with longest prefix

  route packet to the next-hop specified in this entry

end if

Routing in the presence of subnets

- A conventional routing table contains entries like
  - **`<net-addr, next-hop-addr, interface>`**
  - First 3 bits of an IP address informs address class

- In presence of subnets
  - Not possible to know which bits corresponds to the network portion, from the address alone
  - Routing table entries of the form

`<net-addr, subnet-mask, next-hop-addr, interface>`

Forwarding example with subnet masks

Consider the following routing table:

NetworkAddress	SubnetMask	NextHopAddress	Interface
128.96.170.0	255.255.254.0	-	Interface 0
128.96.168.0	255.255.254.0	-	Interface 1
128.96.166.0	255.255.254.0	R2 (*IP Address*)	Interface 2
128.96.164.0	255.255.252.0	R3 (*IP Address*)	Interface 0
Default		R4 (*IP Address*)	Interface 3

Packets with the following destination IP addresses will be sent to which interfaces ?

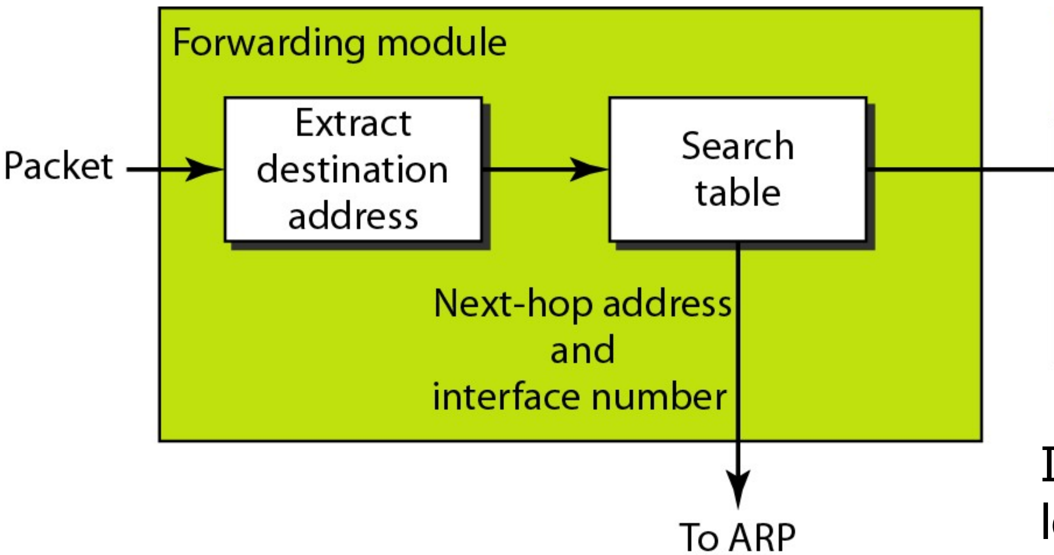
a. 128.96.171.92
b. 128.96.167.151
c. 128.96.163.151
d. 252.5.15.111

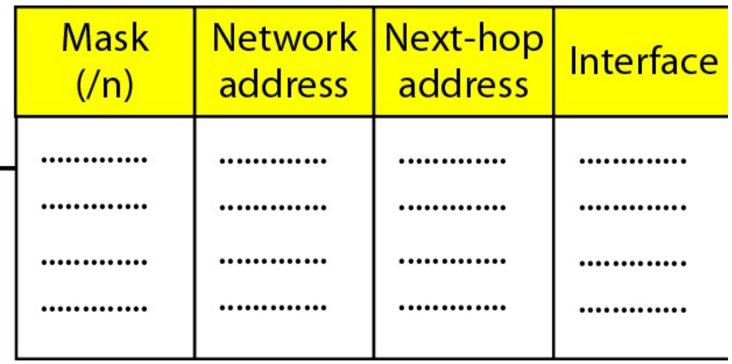
a. Interface 0
b. Interface 2
c. Interface 3
d. Interface 3

Classful addressing routing table

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host-specific	/32	192.16.7.1	202.45.9.3	m0
	/24	193.14.5.0	84.78.4.12	m2
Default	/0	/0	145.11.10.6	m0

Simplified forwarding module in classless address





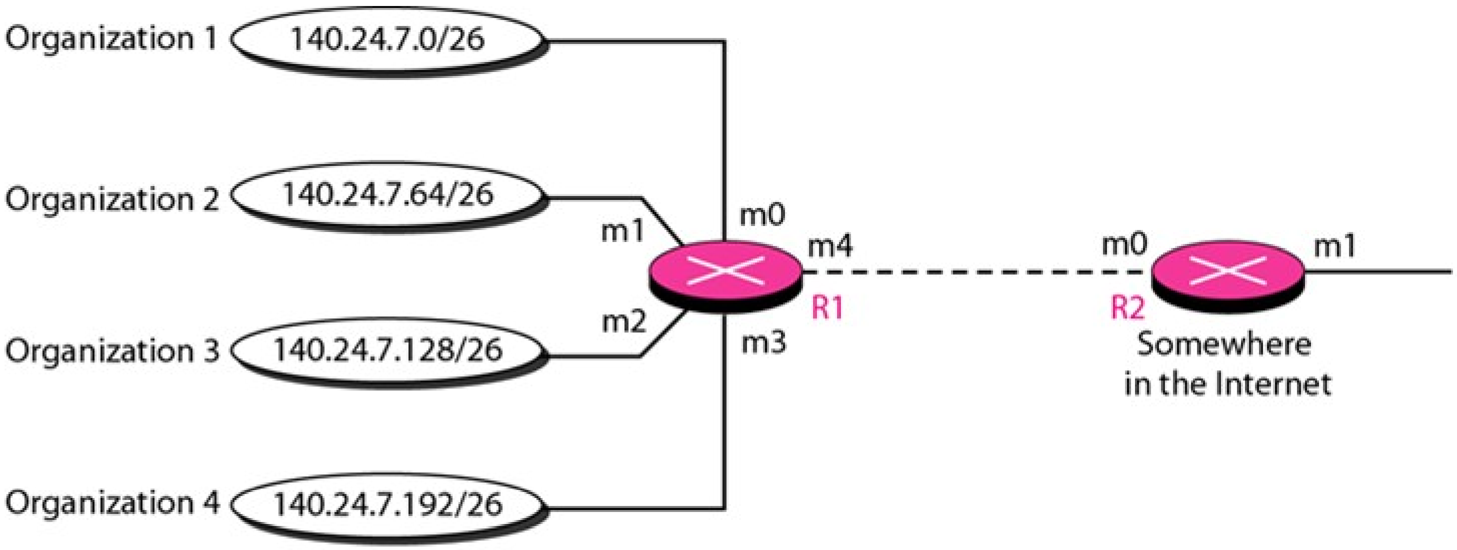
In classless addressing, we need at least four columns in a routing table.

A typical Network Configuration

Routing table for router R1

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Address aggregation



Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	Addr of R2	m4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	Default	m1

Routing table for R2

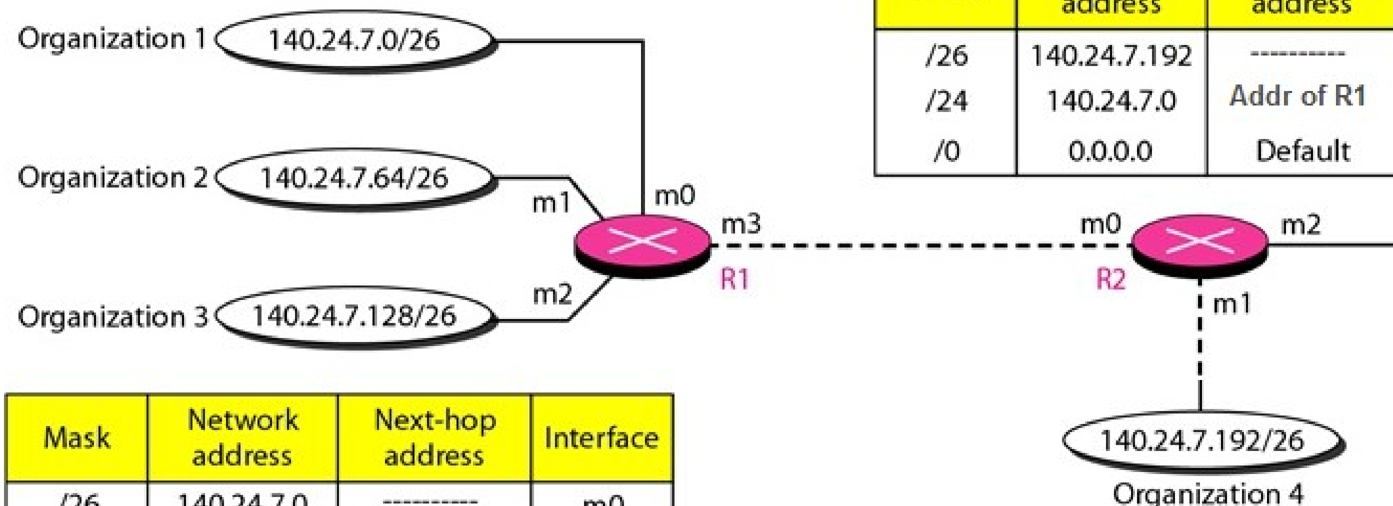
Here four organizations geographically situated close to each other, that’s why this kind of configuration is possible. What happens if they are not ?

Longest mask matching

Suppose a packet arrives at router R2 for org-4 with destination address 14.24.7.200. How it will be routed?

Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m1
/24	140.24.7.0	Addr of R1	m0
/0	0.0.0.0	Default	m2



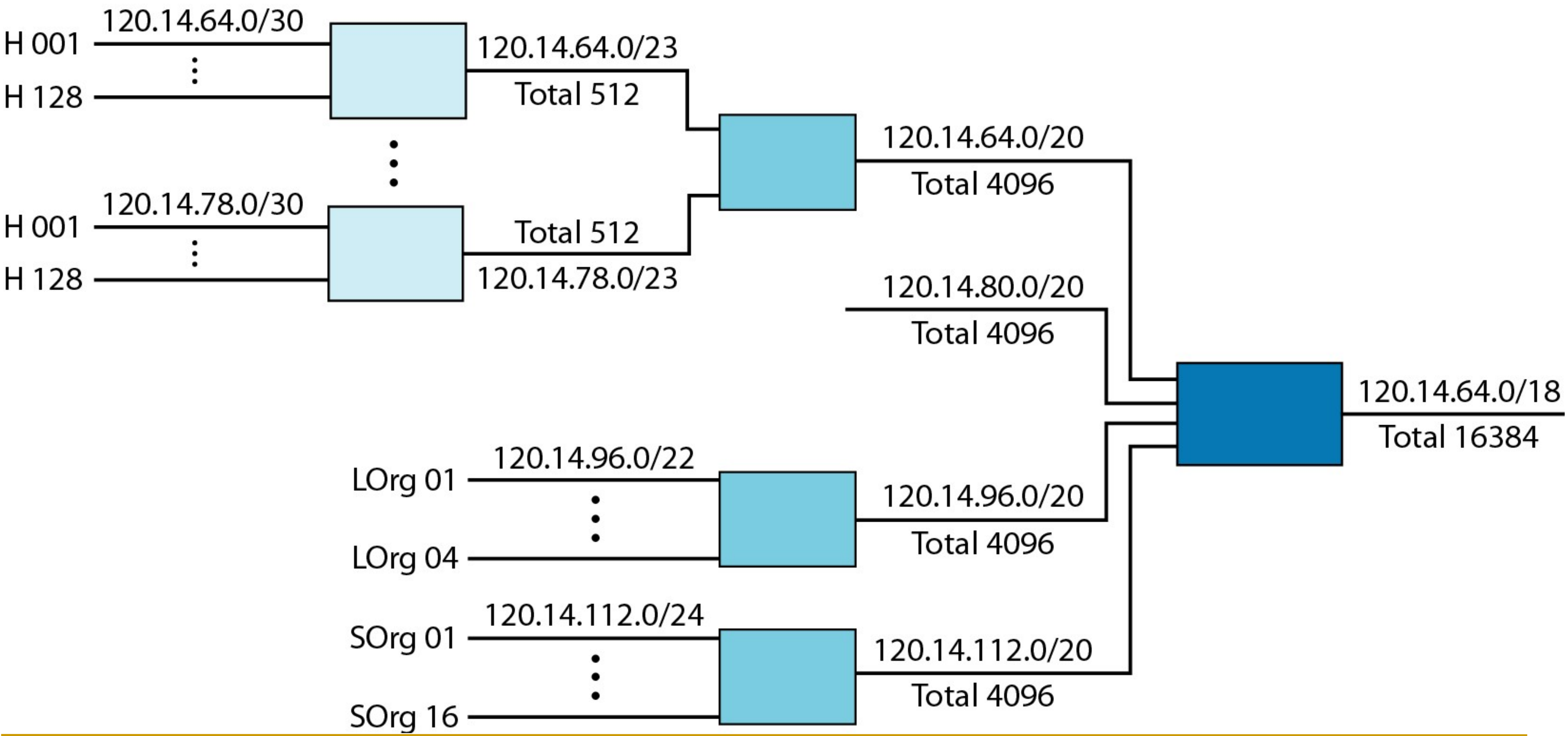
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/0	0.0.0.0	Addr of R2	m3

Routing table for R1

What happens if the forwarding table is not sorted with the longest prefix first ?

Hierarchical routing with ISPs

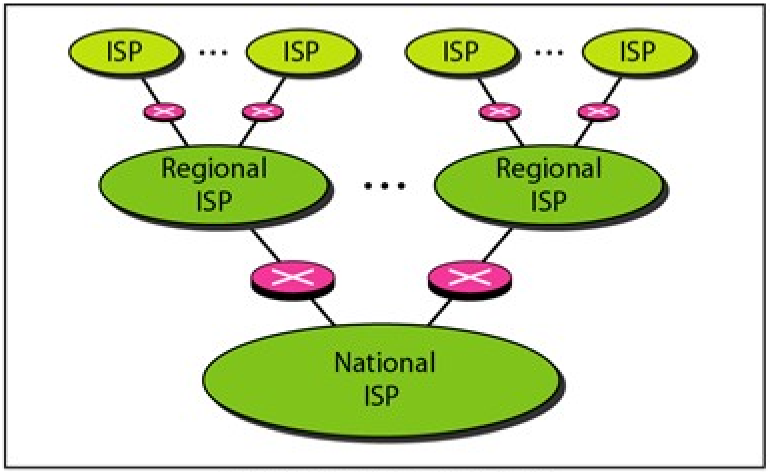
To solve the problem of prolonged routing tables, a sense of hierarchy in the routing tables can be created.



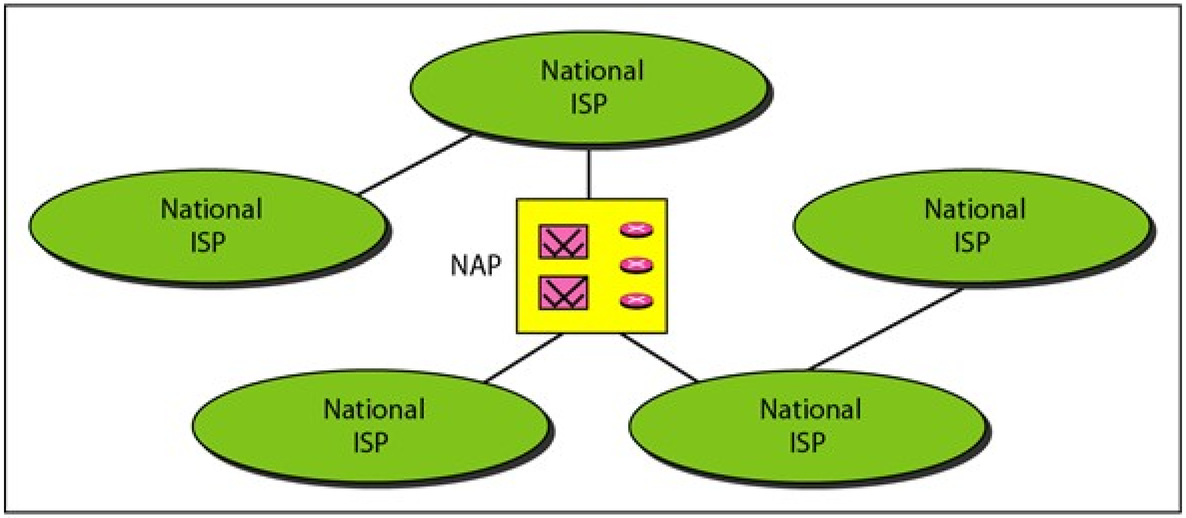
Geographical Routing

- To decrease the size of forwarding table even further, the hierarchical routing is extended to geographical routing
- Entire address space is divided into few large blocks – assign a block to America, a to Europe, a block to Asia and so on
  - The router of ISPs outside Europe will have only one entry for packets to Europe in their forwarding table
  - The routers of ISPs outside North America will have only one entry for packets to North America in their routing tables.
  - And so on.

Hierarchical organization of the Internet



a. Structure of a national ISP

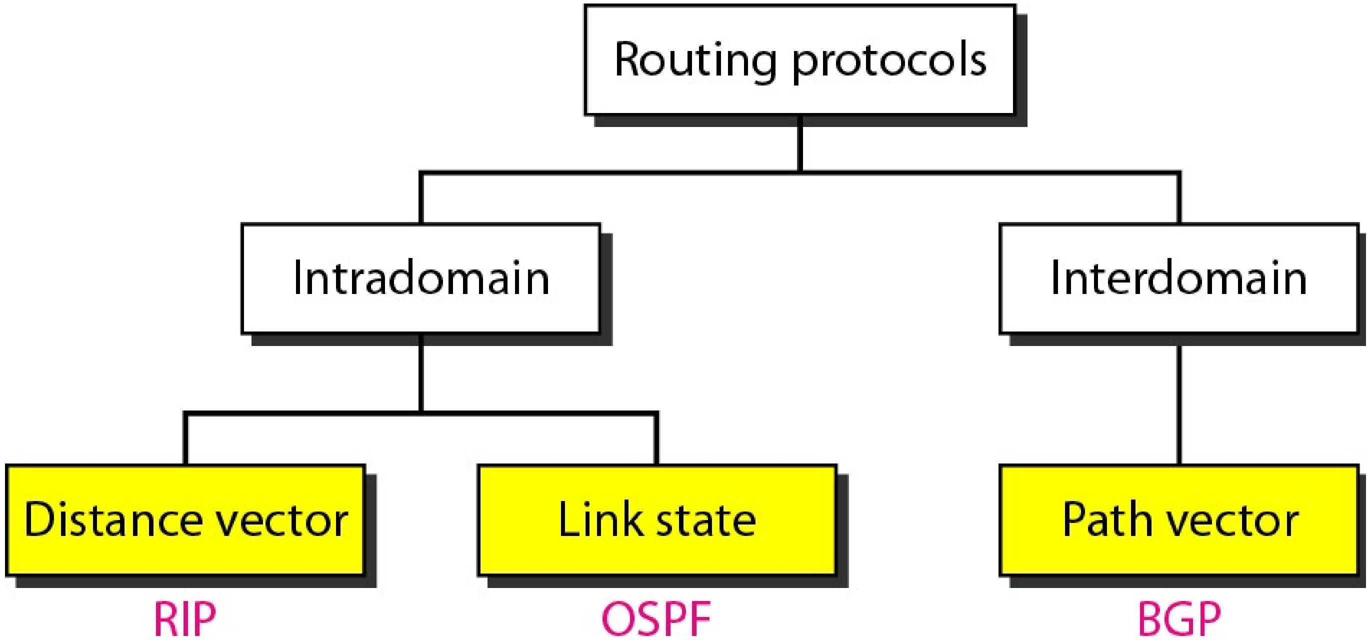


b. Interconnection of national ISPs

The Internet Structure

- Internet has changed from tree-like structure, with a single backbone, to a multi-backbone structure run by different private corporation
- The Internet is so large that one routing protocol cannot handle the task of updating the routing tables of all routers
- Internet is divided into autonomous systems (AS), that is a group of networks and routers under the authority of a single administration (ISP)
  - Each AS can run a routing protocol that meets its need
  - Called *intra-AS routing protocol* or *intradomain routing protocol* or *interior gateway protocol (IGP)*
  - Two common such protocols are RIP and OSPF - AS free to use any
- However the global Internet runs a global protocol to glue all ASs together
  - Called *inter-AS routing protocol* or *interdomain routing protocol* or *exterior gateway protocol (EGP)*
  - Only such protocol - BGP

The Internet-Protocol (IP) Routing protocols



- RIP (Routing Information Protocol)
- OSPF (Open Shortest Path First)
- BGP (Border Gateway Protocol)

Types of Routing

- Fields of a routing table
  - Destination node, next hop, metric, other fields
  - Metric: estimate of the cost of this route to the destination (through the next hop)

- Types of routing protocols
  - Fixed or static
  - Random
  - Flooding
  - Dynamic or adaptive

Fixed / Static routing

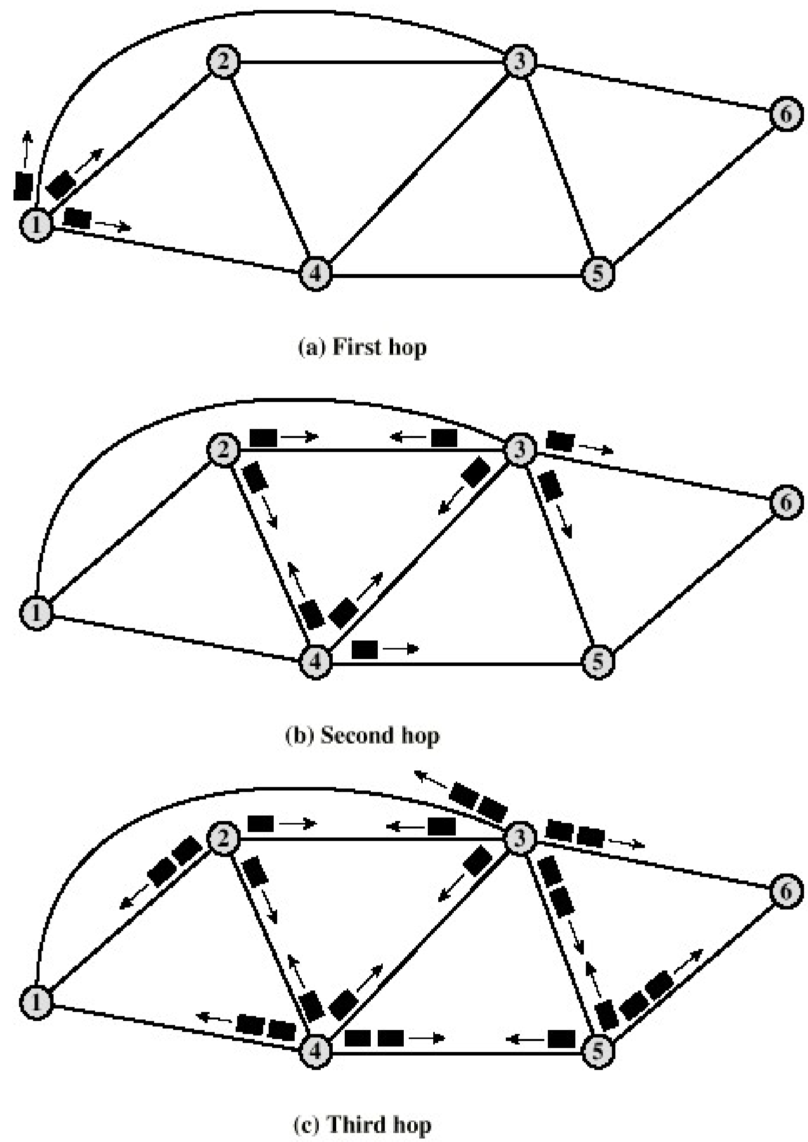
- Single permanent route for each source to destination pair

- Routing tables created & updated manually
  - Routing table is fixed unless manually changed again
  - No dynamic update when network conditions change

- Fine for very small networks

Flooding

- Packets sent by a node to every neighbor, except to the one from which the packet was received
- Eventually a number of copies will arrive at destination
- Each packet is uniquely numbered so duplicates can be discarded



(a) First hop

(b) Second hop

(c) Third hop

Flooding

- Advantages
  - All possible routes are tried, so very robust
  - Can work around failed links/nodes
  - All nodes are visited, so useful to distribute information
  - No network information needs to be stored at any node

- Disadvantages
  - No routes remembered (no routing table)
  - Too many copies of a packet may be sent

Refinements in flooding

- Nodes can remember packets already forwarded to keep network load in bounds

- To handle loops in the network, use hop count

- At least one packet will have taken minimum cost route (e.g. minimum hop count) to reach destination
  - Can be used to set up virtual circuit

- Flooding can be used intermittently, say, when a route to an unknown destination is to be found

Random Routing

- Node selects any one outgoing path for retransmission of incoming packet
  - Selection of outgoing path can be random or round robin or based on probability calculation
- If packet does not reach destination within a time interval, select another outgoing link and transmit
- Advantage
  - No network info needed to be stored at any node, no routes remembered
- Disadvantage
  - Large delay possible, used in networks where delay is not a concern
  - No guarantee that the outgoing path selected will lead to the destination

Adaptive routing

- Routing decisions change dynamically as conditions on the network change
- Storage of info about network at nodes
  - Routes saved in routing tables
  - Routers communicate to update routing tables dynamically when network conditions change

- Advantages
  - No manual intervention necessary, aids congestion control, fault tolerance
- Disadvantages
  - Complex, routing messages are overhead

Issues for adaptive routing protocols

- How much and what network info to collect
- From whom to collect information (from neighbors or from all nodes, etc)
- When to collect information (once at the beginning, or periodically, etc)

- Direct tradeoff between
  - Overhead and
  - Optimality, robustness, speed of convergence

Families of adaptive routing protocols

- Based on with which other nodes does a node exchange routing information
  - Distance Vector routing protocols
  - Link State routing protocols

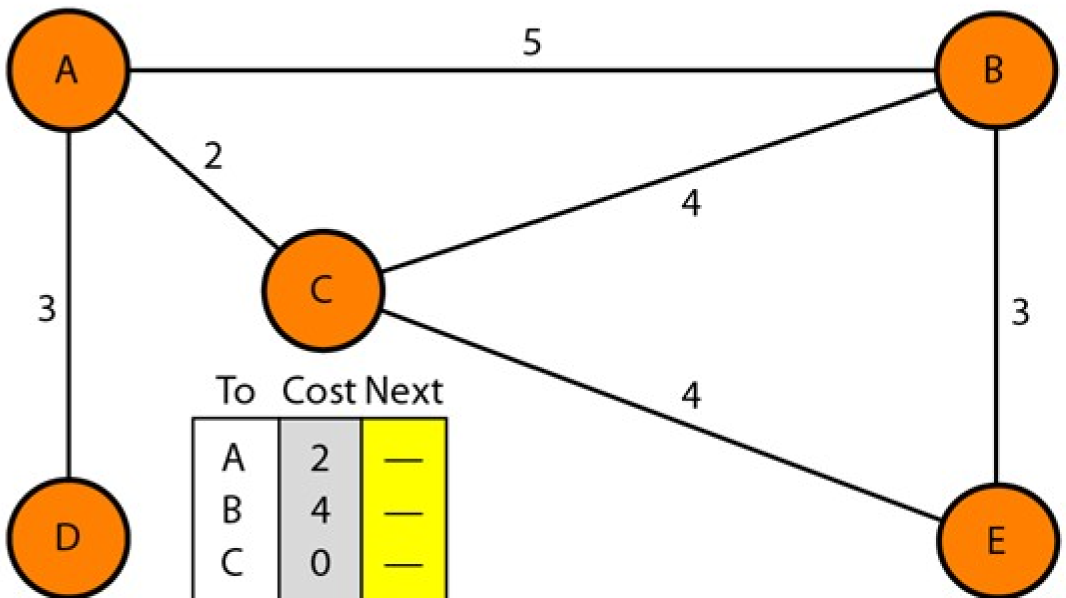
Distance Vector Routing Protocol

Routing Information Protocol (RIP)

Distance Vector Routing

- Each node N keeps track of its least cost path to all other nodes (all nodes that N knows till now)
  - For each node d, N stores the next hop to reach node d (along the least cost path) and the cost of that path
  - This information stored at N, for all nodes other than N, comprises the distance vector at node N
- Distance vector periodically sent to all nbrs, whether or not vector changed since last send
- If no information received from a neighbor within a time, that link is assumed to be down

Initialization of tables in distance vector routing



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	∞	
E	3	—

B's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	∞	
E	4	—

C's table

To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

D's table

To	Cost	Next
A	∞	
B	3	—
C	4	—
D	∞	
E	0	—

E's table

Think nodes as city and links are path connecting them.

Updating in distance vector routing (only a case)

To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's modified table

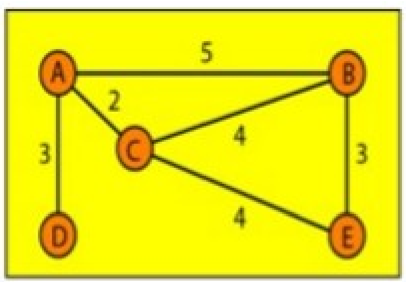
Compare

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's old table

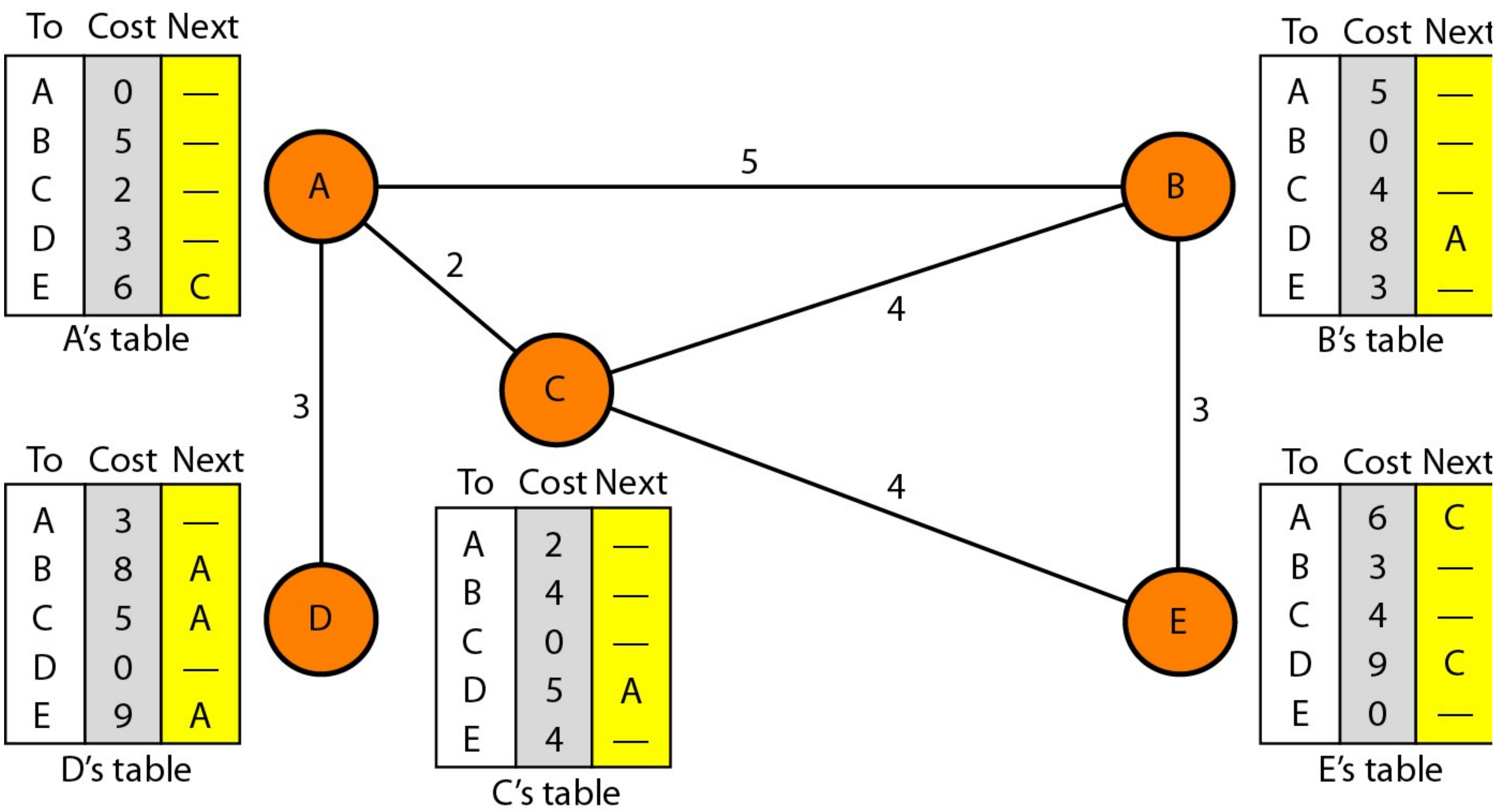
To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's new table



In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

Distance vector routing tables after convergence



The least cost route between any two nodes is the route with min distance

Detection of a node / link crash

- Each node sends its distance vector to all its neighbors periodically

- If nothing received from node N for a certain time, N’s neighbor assumes N has crashed
  - All routes with next hop N are set to have metric INF
  - Routes with metric INF deleted after ‘some’ time defined by actual protocol

- Value of INF defined by the actual protocol being used

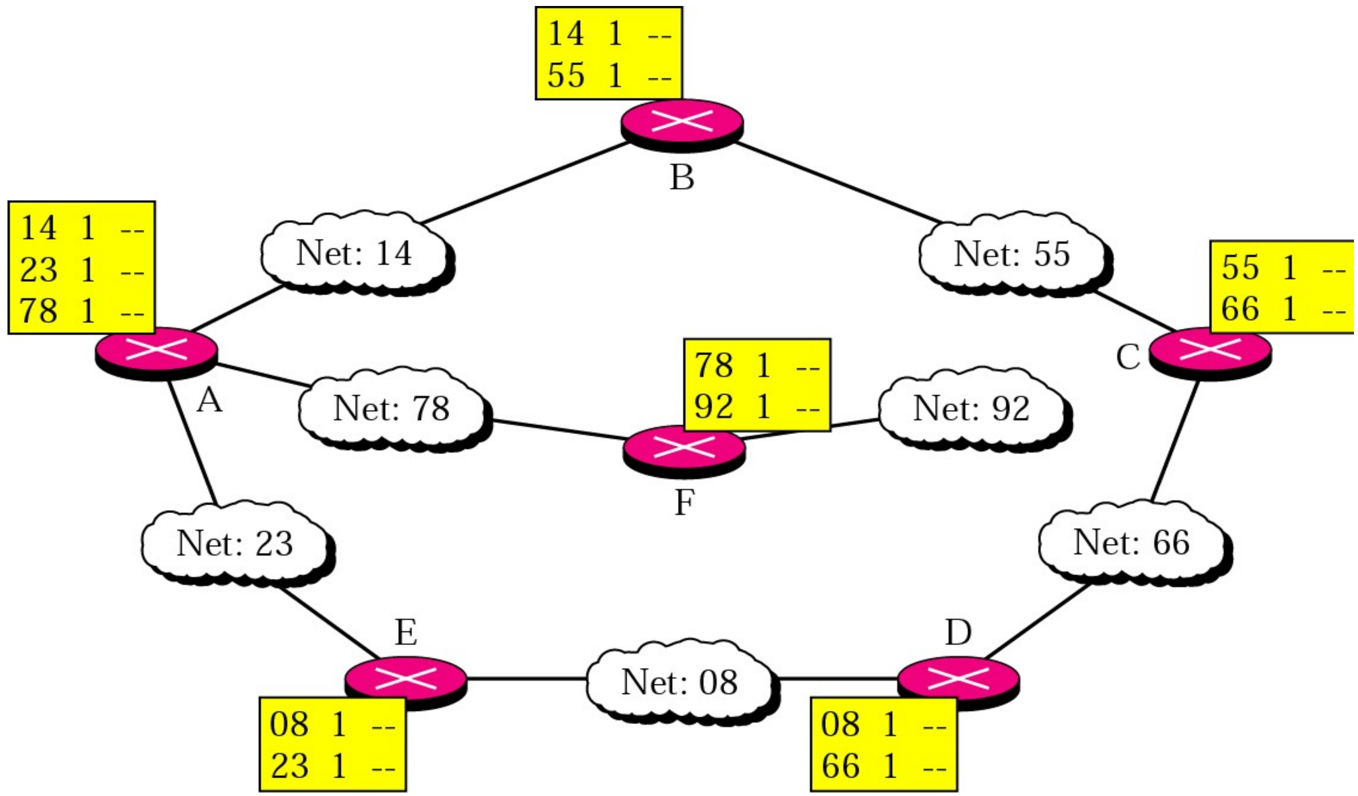
A real-world algorithm based on DVR

- RIP: Routing Information Protocol
  - INF is 16
  - Cost of each link is constant = 1 (as if all links have same cost)
  - Broadcast interval: 30 seconds
  - Route expiry time: 180 seconds
    - ✓ If no message received from a neighbor n for 180 sec, change metric to INF for all routes through n
  - Time to delete a route: 120 sec after metric is set to INF
  - Uses split horizon, hold down, triggered update, split horizon with poissoned reverse (*to be discussed later*)

On receipt of distance vector of node B, at node A in RIP

```
for every entry <d, x, k> in B's distance vector
    if <d, _, _> is not in A's routing table
        add <d, B, k+1> to A's routing table
    else if <d, y, k'> exists in A's routing table
        if k+1 < k'
            replace <d, y, k'> with <d, B, k+1> in A's table
        else if y = B        # trust your next hop
            replace <d,B,k'> with <d,B,k+1> in A's table
        end if
    end if
end for
```

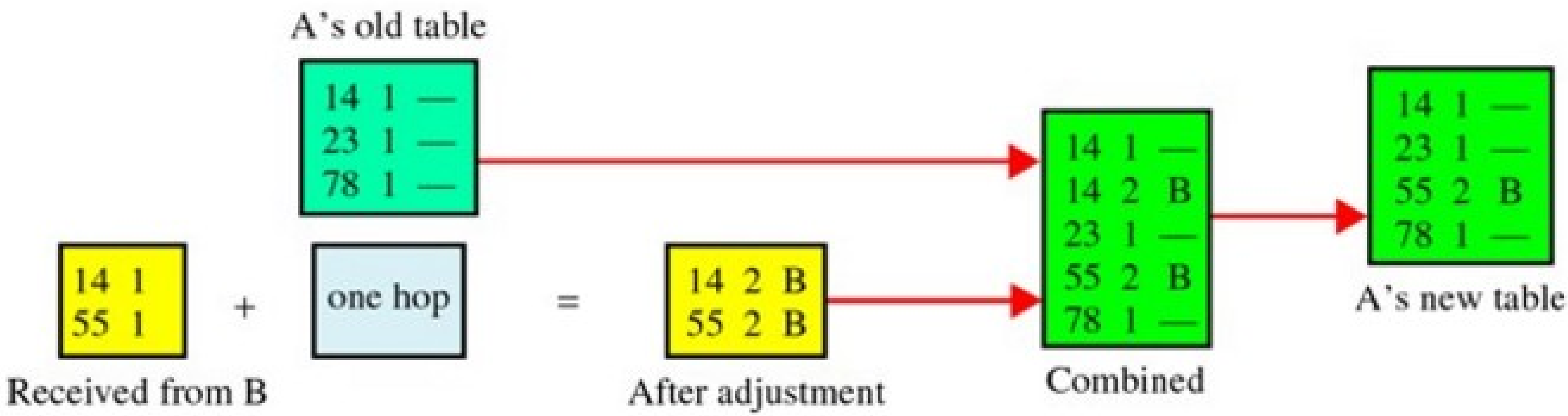
Initial routing tables in an internetwork



- ❑ Each Router initializes a routing table for itself using config files or static routes
- ❑ Table consists only directly attached networks with the hop count set as 1

Note : The next-hop field, which identifies the next router is empty

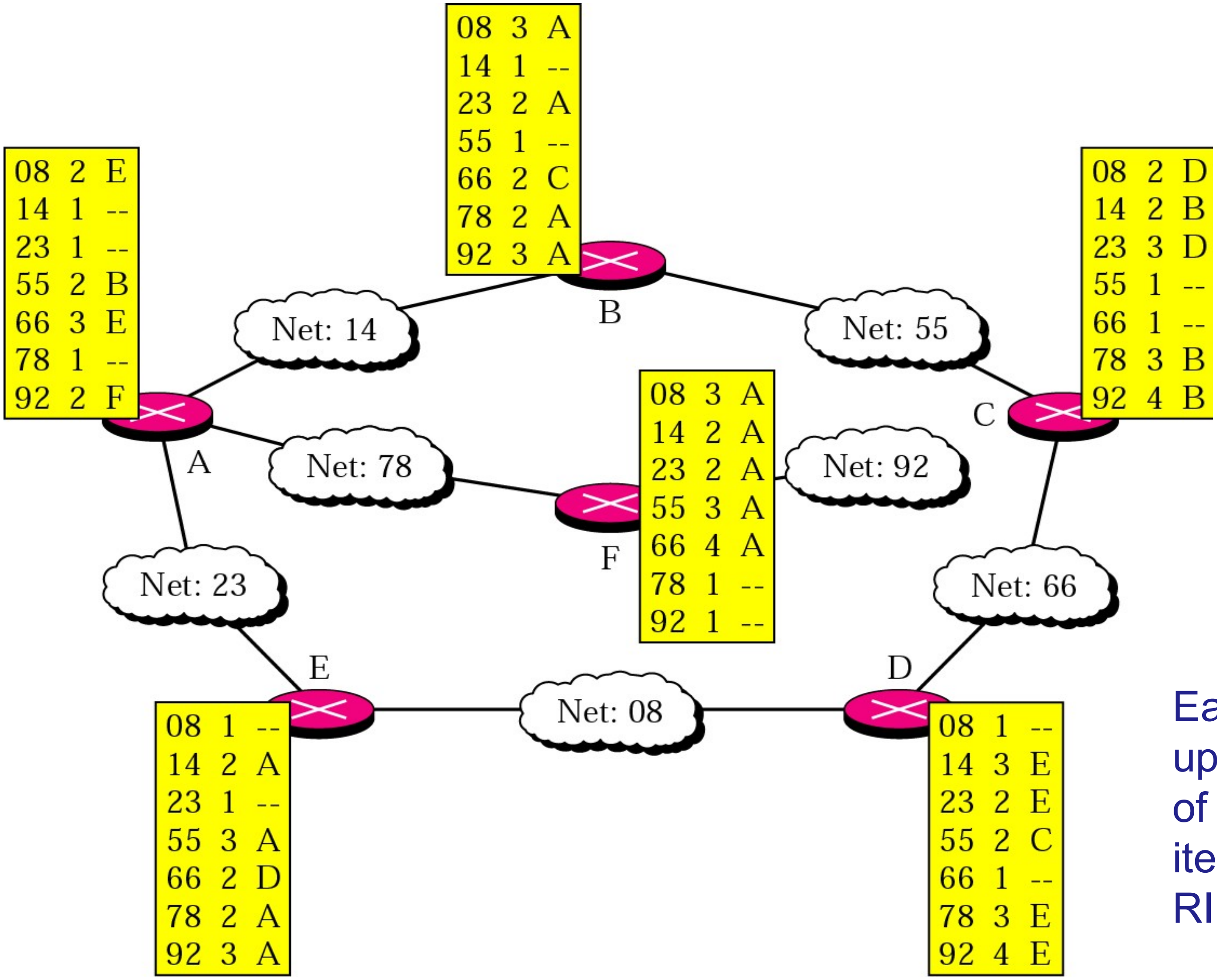
Updating routing tables of Router A



- Updating routing table of node A upon receipt RIP message from Node B

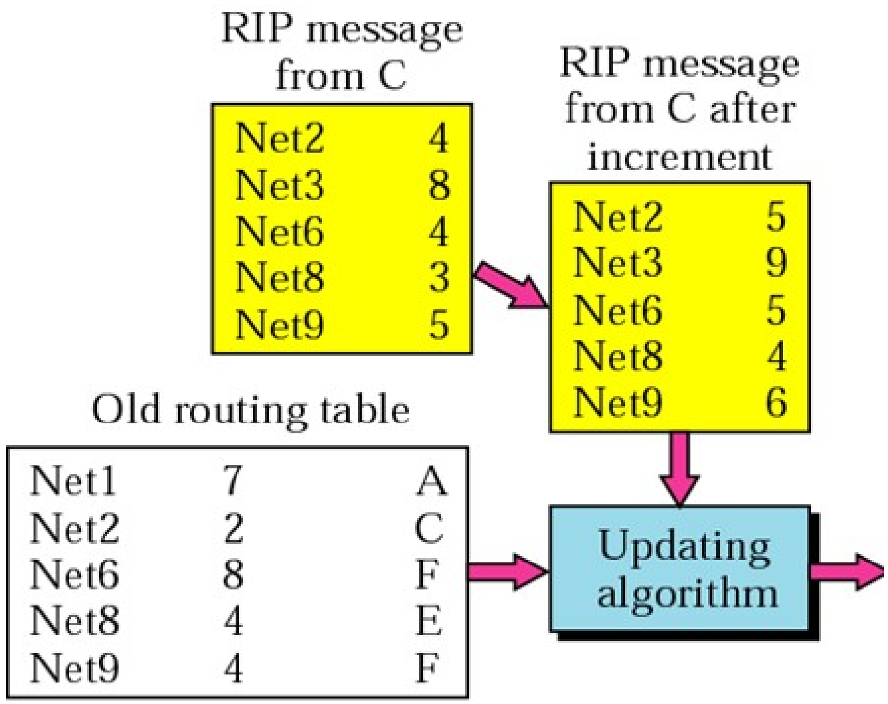
Note : No need to send next hop while sending distance vectors

Final routing tables in the internetwork



Each routing table is updated upon receipt of RIP messages (few iterations) using the RIP updating algorithm

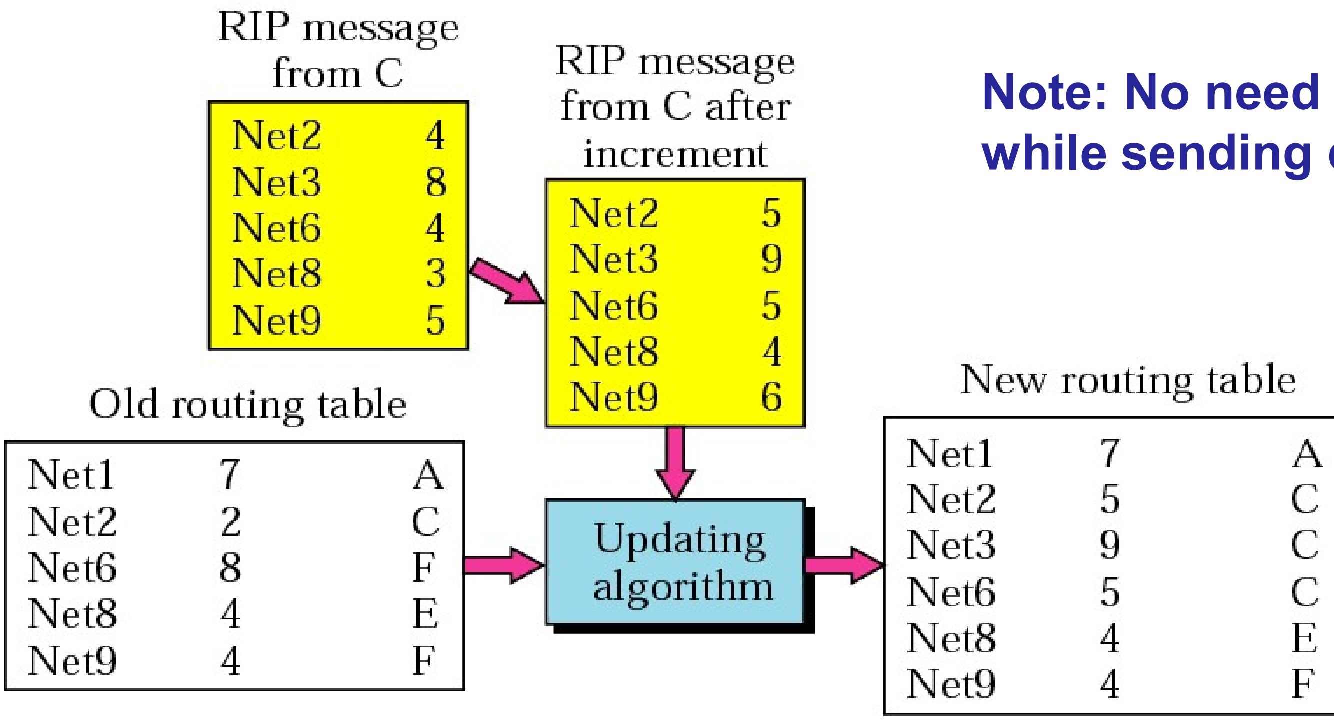
Example of updating a routing table (detail)



Net1: No news
Net2: Same next hop
Net3: A new router
Net6: Different next hop, new hop count smaller
Net8: Different next hop, new hop count the same
Net9: Different next hop, new hop count larger

Note: No need to send next hop while sending distance vectors

Example of updating a routing table (detail)



Note: No need to send next hop while sending distance vectors

Net1: No news, do not change
Net2: Same next hop, replace
Net3: A new router, add
Net6: Different next hop, new hop count smaller, replace
Net8: Different next hop, new hop count the same, do not change
Net9: Different next hop, new hop count larger, do not change

Problems with DVR

- Slow convergence
  - If a part of the network becomes inaccessible, it may take a long time for all other nodes to know this
- Too much overhead – updates sent periodically even if no change in routing table
- Routing loops may take a long time to be detected (count to infinity problem)

Routing loop or Count-to-Infinity problem

A ——— B ——— C

	B	C
	A, A, 1	A, B, 2
B detects A has crashed	A, A, Inf	A, B, 2

- If B’s routing table reaches C first, no problems
  - B sends <A, Inf> to C, so C updates to <A, B, Inf>

- But what if C’s routing table reaches B first?

Slow Convergence and Count-to-Infinity problem

- Although it converges to the correct answer, it may do so slowly. In particular,
  - It reacts rapidly to good news, but
  - leisurely to bad news.

A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

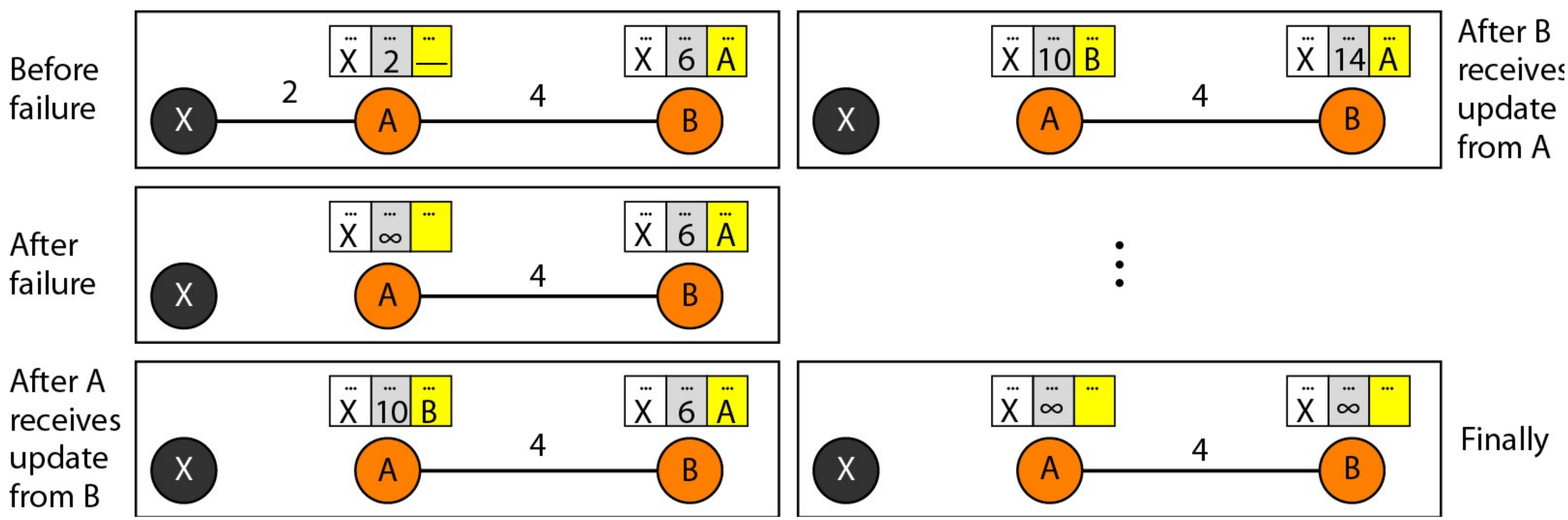
(a) Response to good news –
Node A is down initially and all the other routers know this

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b) Response to bad news -
Suddenly, either A goes down or the link between A and B is cut

Two-node Loop Instability

A problem with distance vector routing is instability, which means that a network using this protocol can become unstable



Packets bounce between A and B, creating a two-node loop problem

To avoid count-to-infinity

- Split horizon technique
  - Do not send a route to neighbor X, if the next hop in this route is X itself
  - If network has a loop of length > 2, problem may occur even if split horizon technique used (*three node instability*)
- Triggered updates
  - If a metric changes to INF, send it to all neighbors immediately
- Hold down
  - if a metric has changed to INF, do not change it to a lower value for some time
- Split horizon with poissoned reverse
  - Send all entries to neighbor X, but advertise the metric as INF if next hop in a route is X

All these techniques decrease the probability of formation of routing loops, but they can still occur

Link State Routing Protocol

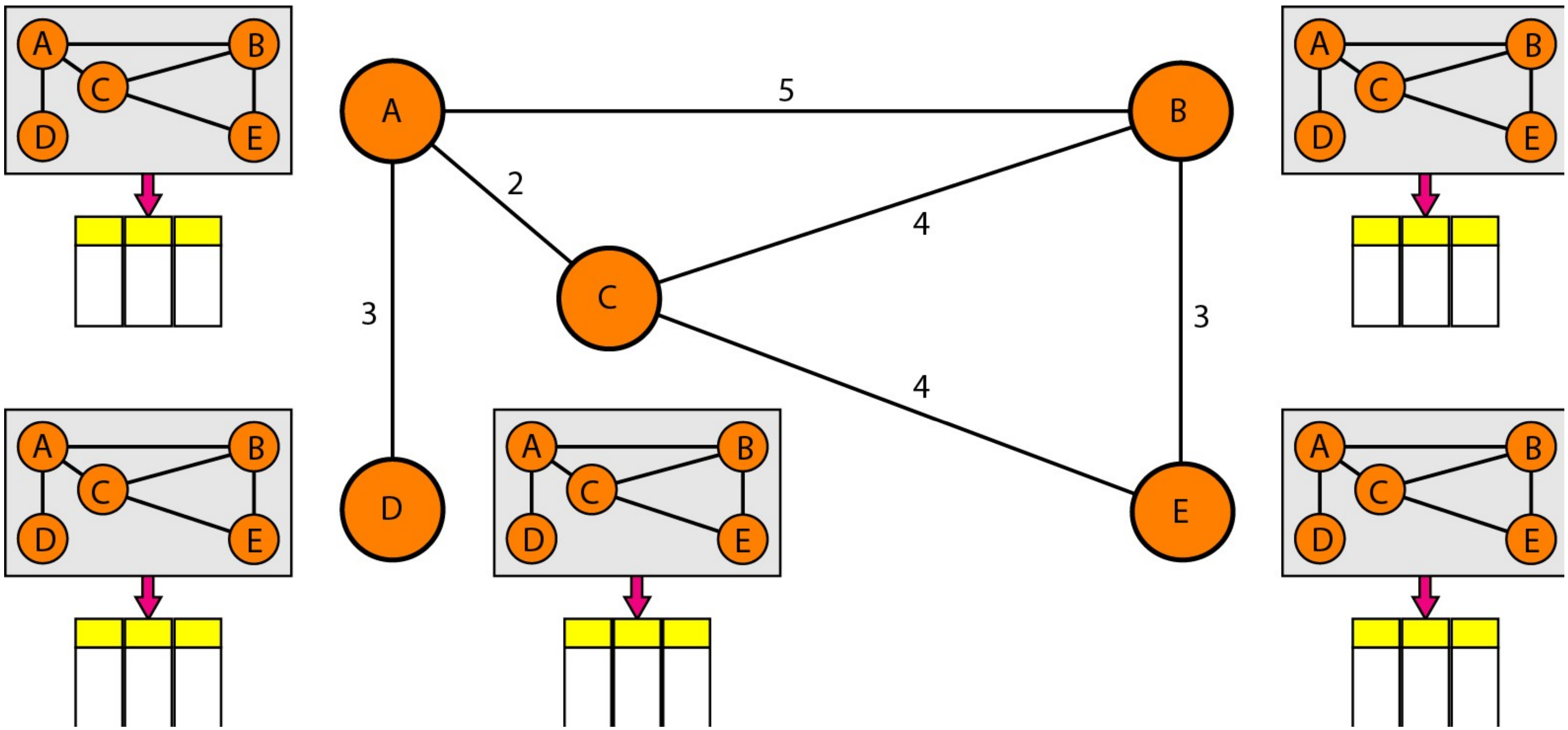
Open Shortest Path First (OSPF)

Link State Routing protocols

- Each node sends only information about its neighbors (who and cost) to **all nodes** in the network

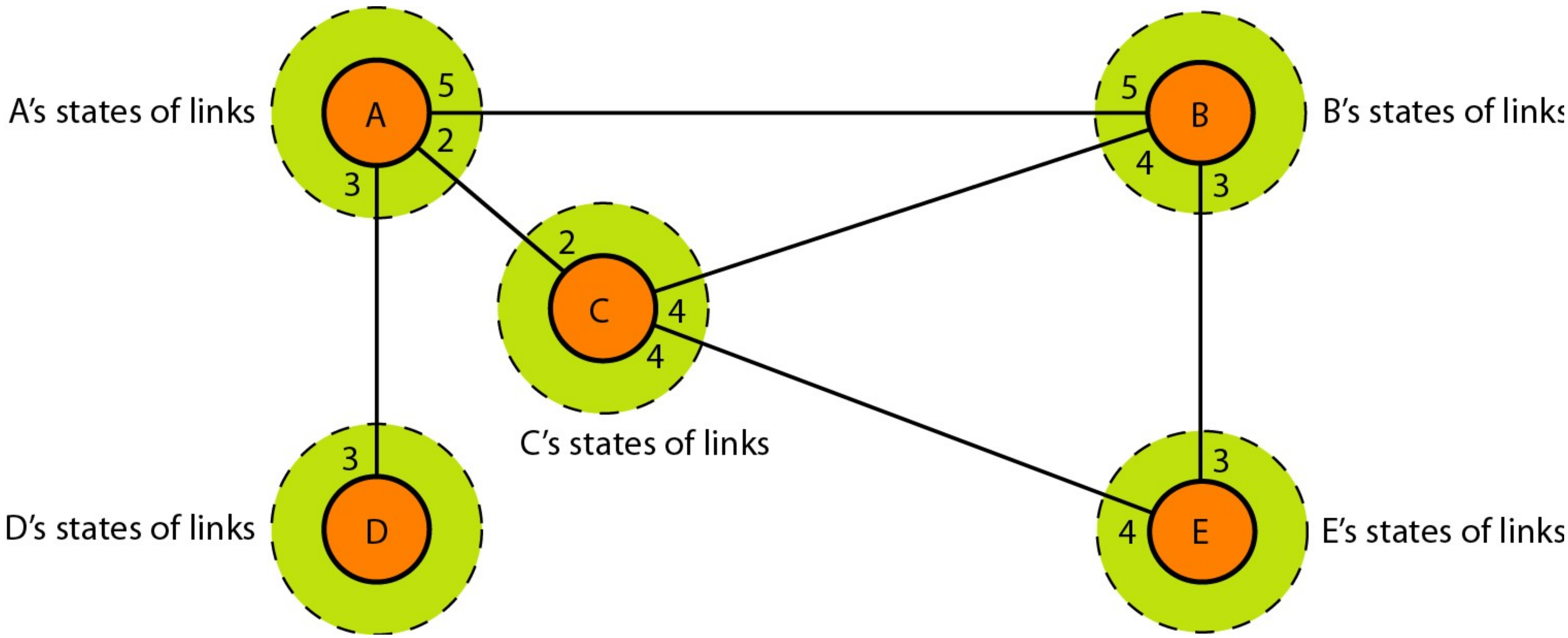
- This information sent to all nodes by Flooding whenever a change is detected (e.g. a broken link, a crashed neighbor) among neighbors
  - Standard flooding optimization techniques used
  - Neighborhood information also sent periodically, along with if change is detected
  - This period can be relatively much larger compared to broadcast interval of DVR

Concept of link state routing

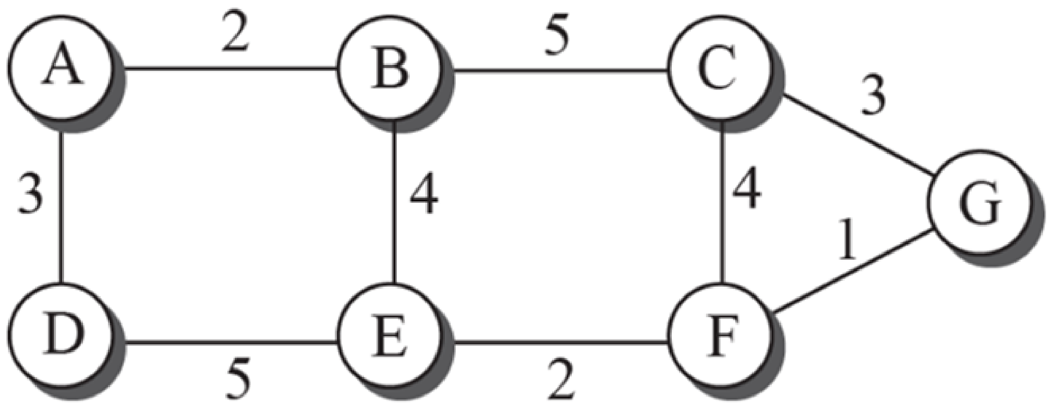


- Each node needs to have a complete map of the network, which means it needs to know the state of the each link
- Collections of states for all links is called *Link-state Database* (LSDB)

Link state knowledge



Example of Link State Database



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

Link state database

Steps in Link State Routing

1. Discover your neighbors
2. Measure the delay to your neighbor
3. Build the Link State Packet (LSP)
4. Distribute the LSP to all nodes
5. Calculate the shortest path to all other nodes

Steps of LS Routing (contd.)

1. Discover your neighbors
   - As a router comes up, sends a ‘hello’ packet (contains own IP address) on all outgoing links
   - Gets the reply (from all neighbors) with “who it is” information e.g. IP address of neighbor

2. Measure the delay to your neighbor
   - Send out an ‘echo’ packet
   - neighbor sends it back immediately
   - (Round trip time / 2) gives the delay to the neighbor
   - Repeated several times to get the average delay

Steps of LS Routing: Build the LSP

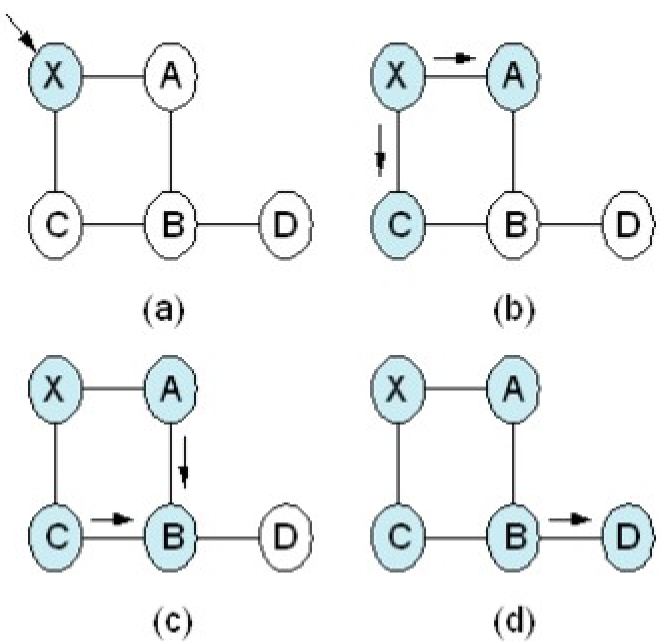
3. The LSP sent by node N contains
   - The identification of the generating router N
   - Sequence number
   - Age / time to live (ttl)
   - List of identifications and distances to the neighbors

- Sequence number in LSP
  - Ensures that no older version of the LSP (sent previously by N) is used by other nodes
  - A node sends LSPs with continually increasing sequence numbers

Steps of LS Routing (contd.)

4. Distribute the LSP by flooding

- Flooding optimizations used
  - LSP not sent to the neighbor from whom it is obtained
  - A router M receives a LSP from router N



- ✓ Holds LSP for some time period T, to see if some other LSP from the same source N is coming
- ✓ After this time T, decides whether or not to flood the LSP
- ✓ If multiple copies of LSP (with same sequence no. and from same source) got within time T, flood one copy only
- ✓ If a LSP with higher sequence number received from same source within time T, discard older LSP, flood new LSP

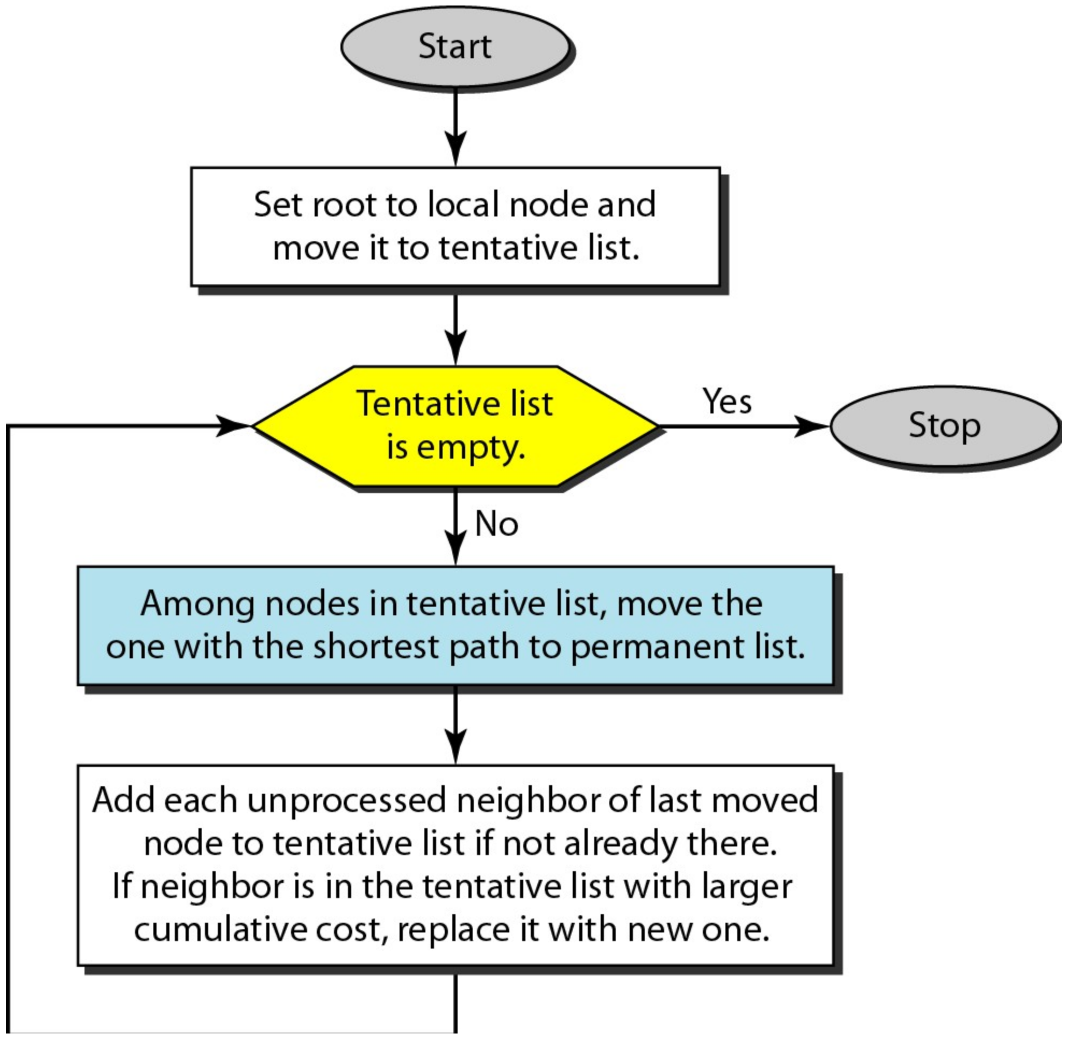
Steps of LS routing: Compute shortest path

5. Once a node has received LSPs from all (most) other nodes
   - it knows the entire network (or a large part of it)
   - Uses Dijkstra's algorithm locally to compute shortest path to all nodes
   - Routing table (next hop and cost of path to reach each node) obtained from output of Dijkstra's algo

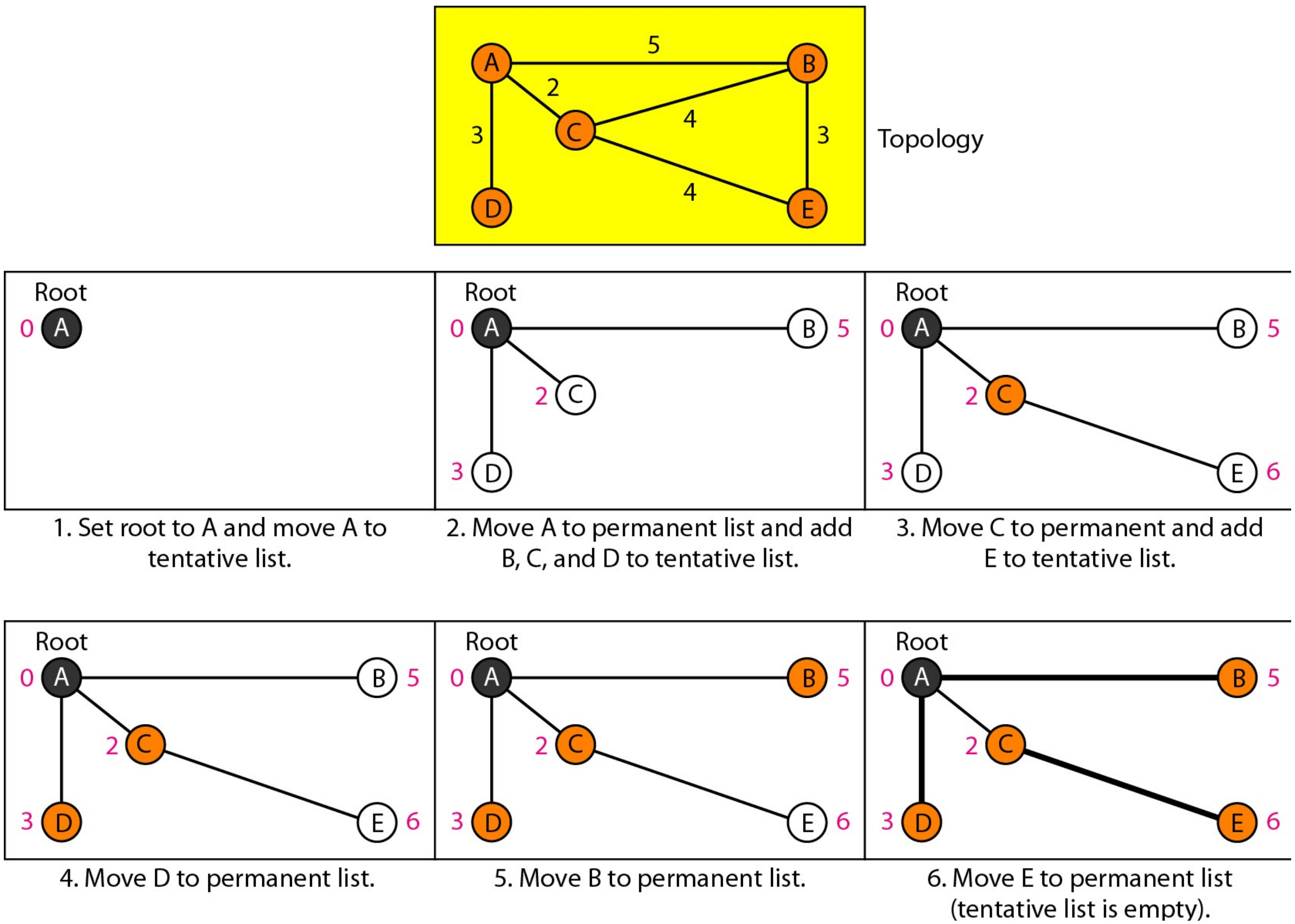
Dijkstra algorithm

- Calculates the shortest path between two points on a network, using a graph made up of nodes and edges.

- Algorithm divides the nodes into two sets: Tentative and permanent. It chooses nodes, makes them tentative, examines them, and if they pass the criteria, makes them permanent.



Example of formation of shortest path tree



Routing table for node A

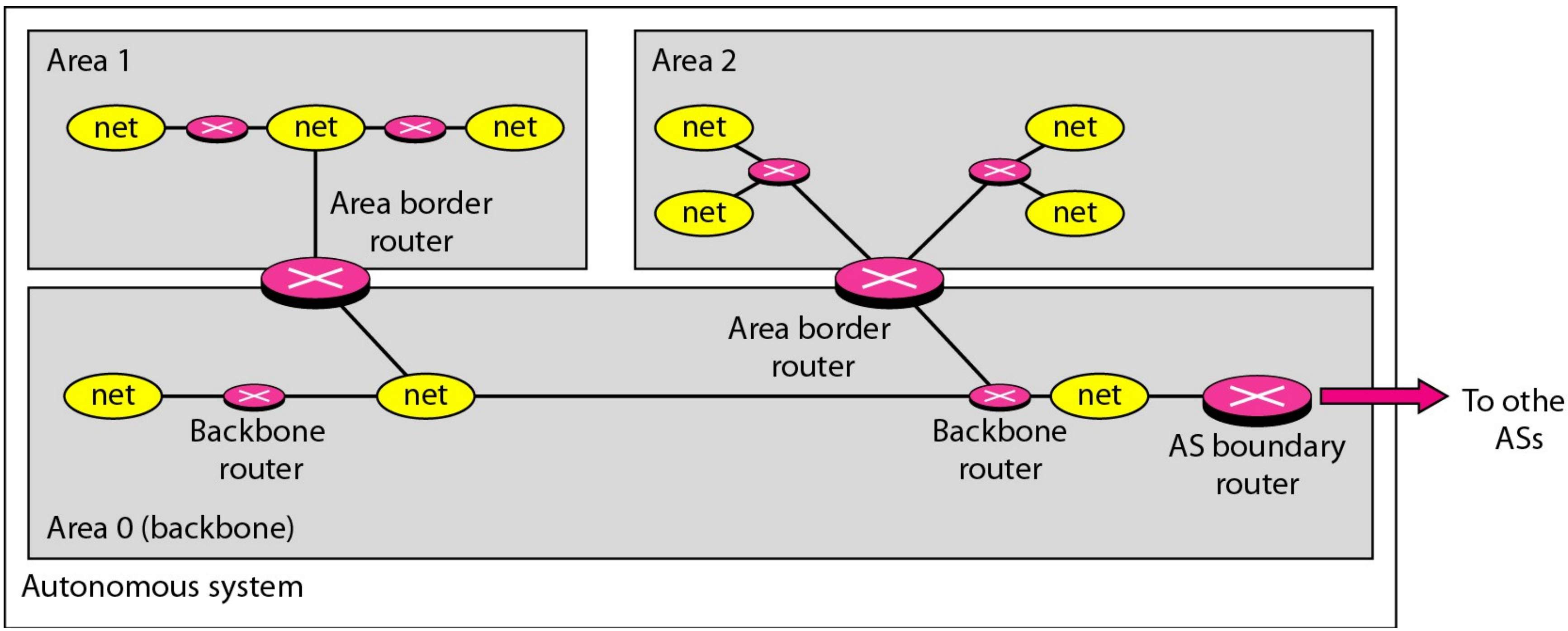
Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A practical algorithm based on LS

- OSPF (Open Shortest Path First)
  - Period of flooding: 30 minutes
  - Metric used can be flexibly defined to indicate link cost, etc
  - OSPF keeps multiple (if more than one exist) least-cost routes to reach same destination
    - ✓ This allows some load balancing

Areas in an autonomous system

- OSPF may not create problem in small AS, but may create problem in larger network – all router flood the whole AS with their LSP
- Therefore, OSPF divides an autonomous system into areas.
- Special routers called autonomous system boundary routers, are responsible for dissipating information about other autonomous system into the current system



Path Vector Routing Protocol

Border Gateway Protocol (BGP)

For Further Studies

References

- *Data Communications & Networking, 5th Edition, Behrouz A. Forouzan*
- *Computer Networks, Andrew S. Tanenbaum and David J. Wetherall*
- *Wikipedia*

Appendix