Computer Networks and Communication

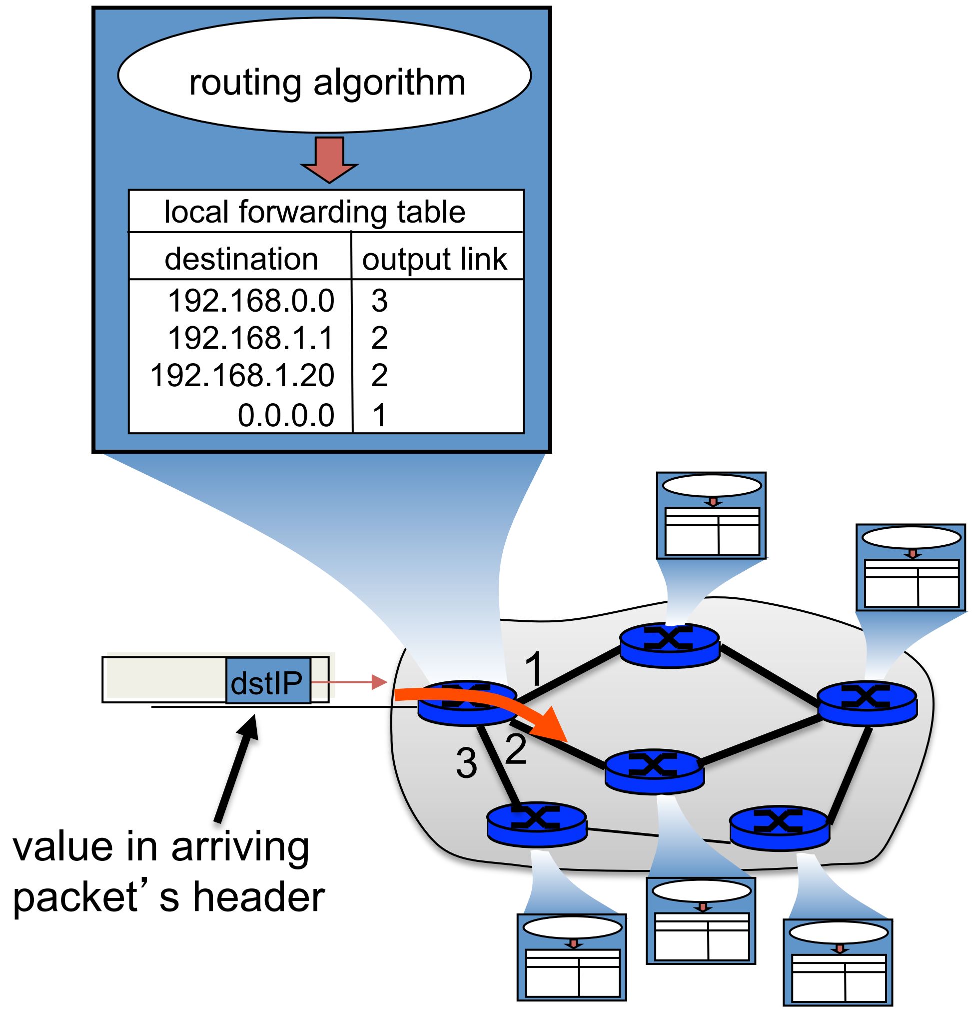
Lecture 9

Network Layer II:

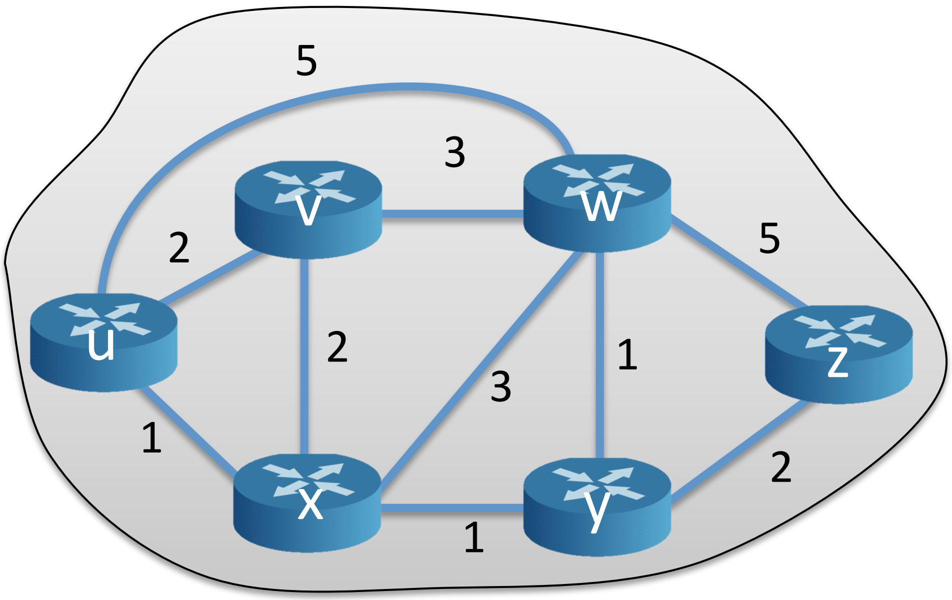
Routing Algorithms

Packet Routing and Forwarding

- A router forwards a packet through a link based on its **forwarding table**
- The forwarding table is created by a **routing algorithm**



Graph Abstraction of a Topology

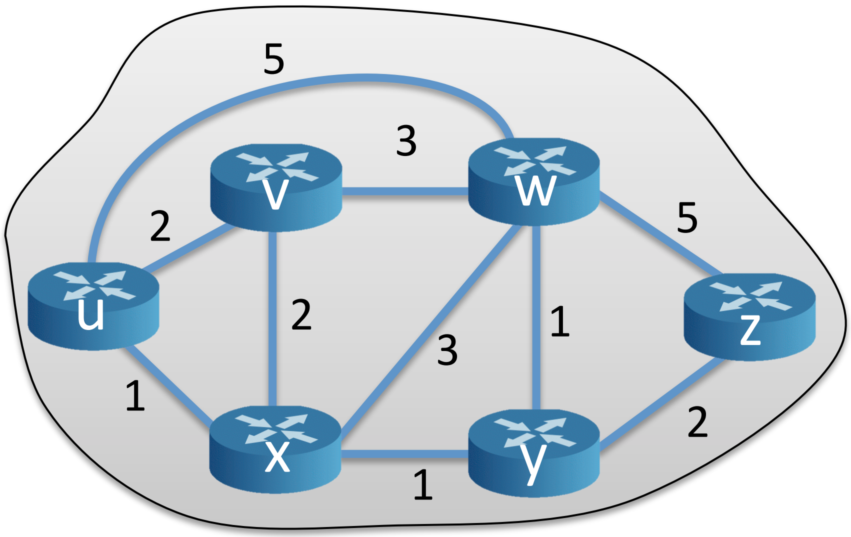


Graph: $G = (N,E)$

N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Cost of a Path



- $c(x,x')$ = cost of link (x,x')
  - e.g., $c(w,z) = 5$
- Cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

Cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Question: What's the least-cost path between u and z ?

Cost of a Path (2)

- What could be used as the **cost** of a path?
  - That is what could be used as the function c(x,x')?
  - Distance
  - Bandwidth
  - Average traffic
  - Number of hops
- Should the cost from x to x' and x' to x be equal?
  - That is, should c(x,x') = c(x',x)?
  - In which case would c(x,x') ≠ c(x',x)?
- If all links have the same cost, the **least-cost path** is also the **shortest path**

Routing Algorithm

- **Routing algorithm** is a piece of software that find the least-cost path from one network node to another
- They can be divided into two categories
  - **Global information**
    - All routers have complete network structure (topology)
    - **Link state algorithms**
  - **Decentralized information**
    - Each router has knowledge of adjacent routers only
    - **Distance vector algorithms**

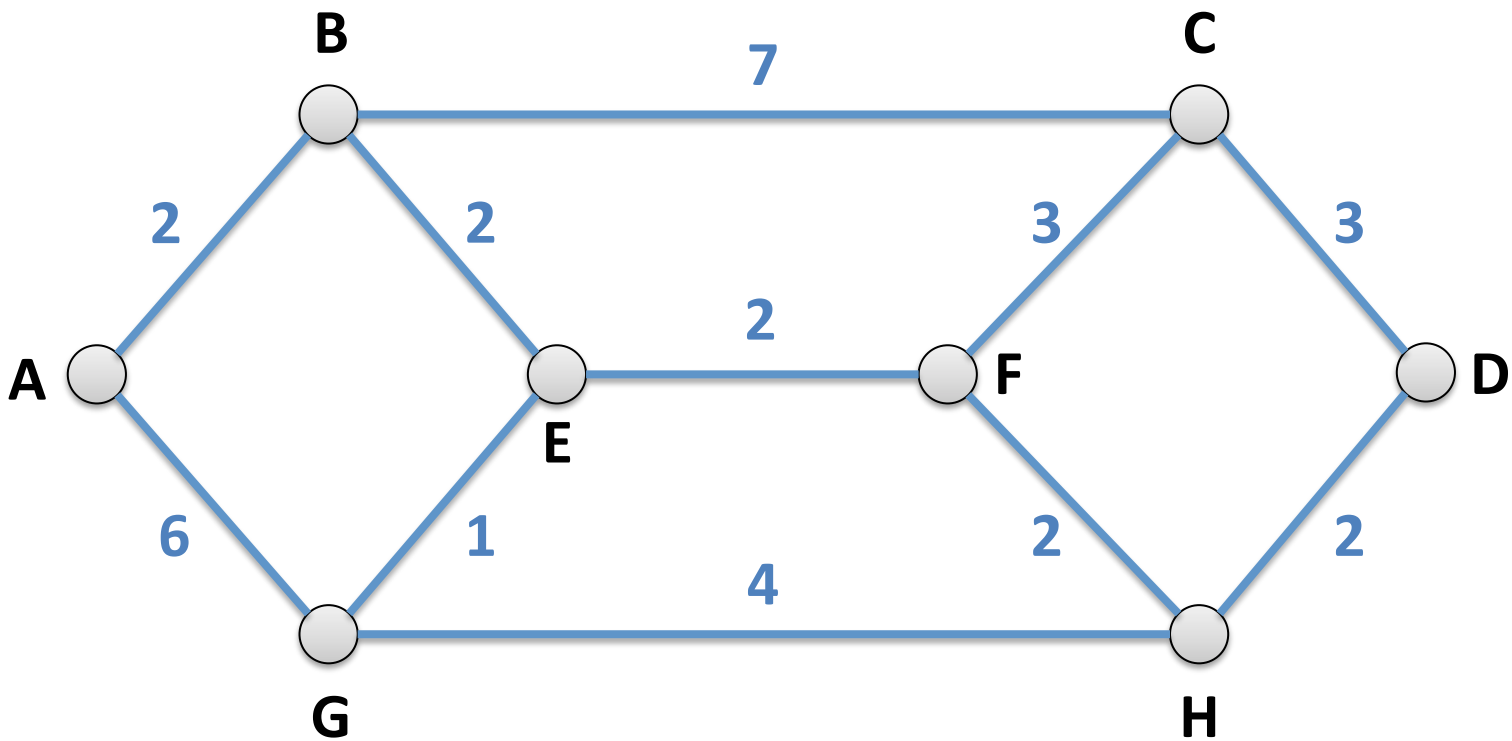
A Link-State Routing Algorithm

- **Dijkstra's algorithm**
  - Compute the least-cost path from a node x to x'
  - Topology and link costs are known to all nodes
    - Such information is shared using link-state broadcast
    - All nodes have the same information
  - Computes least-cost paths from one node (the source) to all other nodes
  - Introduced by Dutch mathematician Edsger Dijkstra in 1959

Dijkstra's Algorithm

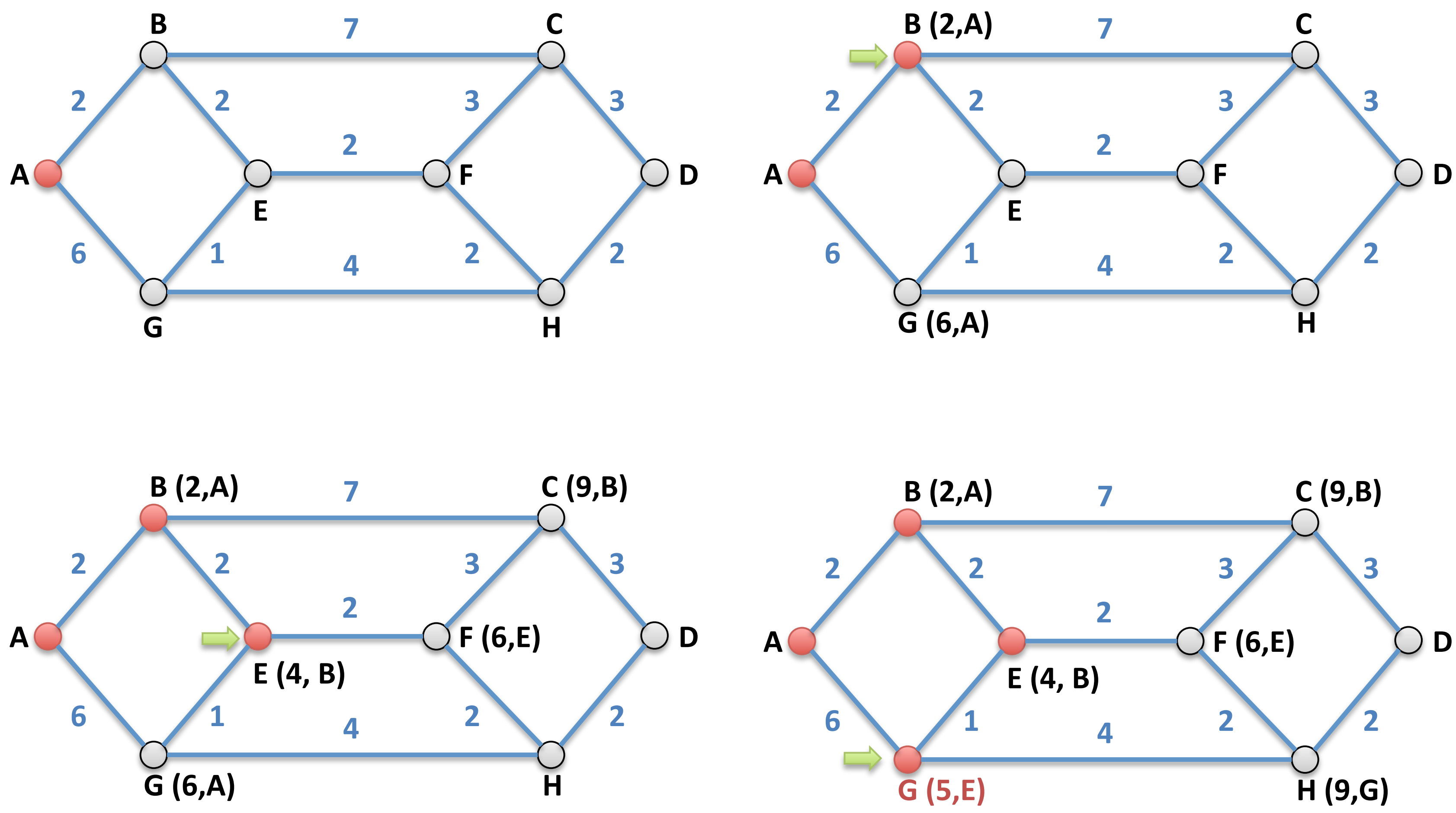
- Initially, assign a distance value (cost) to every node
  - Cost of initial node = 1
  - Cost of other nodes = infinity
- Mark all nodes unvisited. Mark initial node as **current**
- For **current node**, consider all unvisited neighbors and calculate their **tentative** distance
- If all neighbors of the current node is considered, mark it as **visited**
- Move **current node** to the unvisited node with the lowest distance
- If all nodes have been visited, the algorithm terminates.

Dijkstra's Algorithm in Action



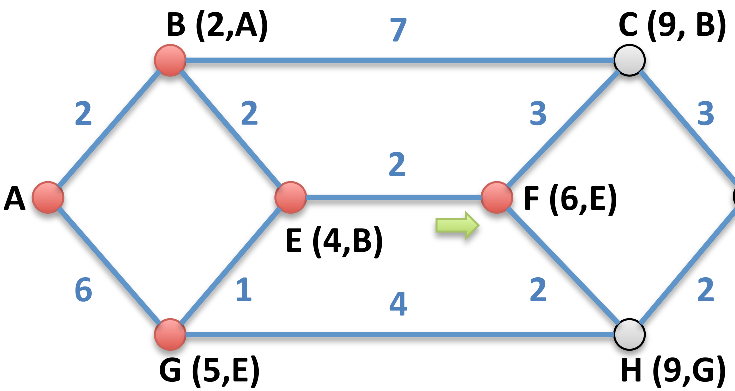
Find a shortest path from A to D

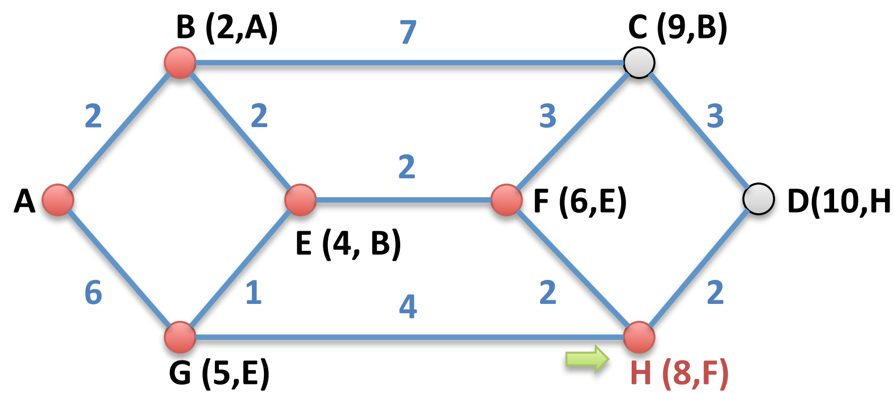
Dijkstra's Algorithm in Action (2)

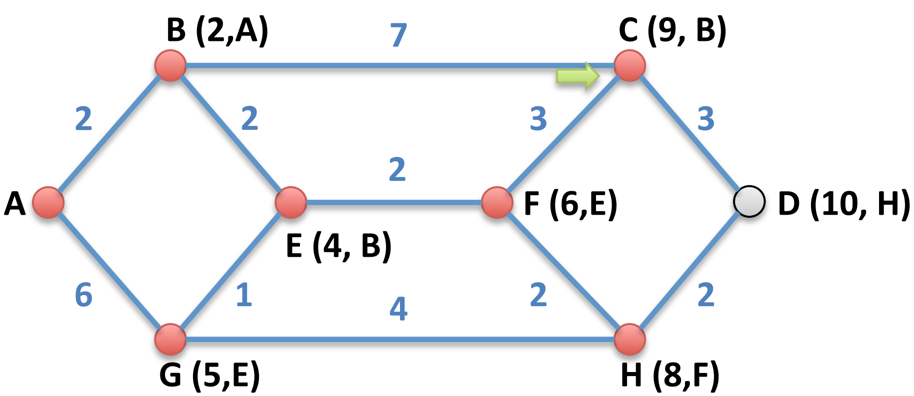


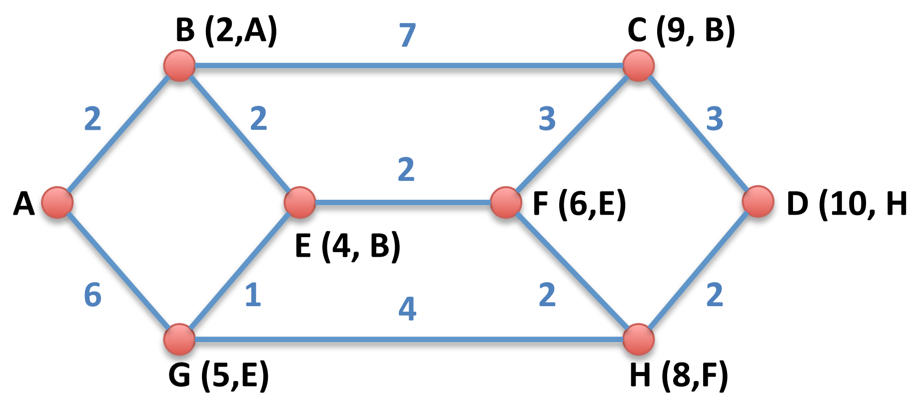
Note that G's cost is updated

Dijkstra's Algorithm in Action (3)









Shortest / least-cost path: ABEFHD

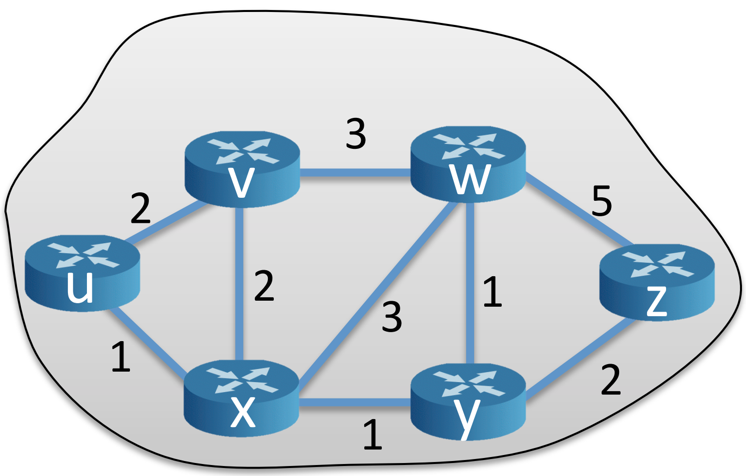
A Distance Vector Algorithm

- **Bellman-Ford algorithm**
- Each router maintains a table (i.e. a **vector**) giving the least-cost path to all other nodes
- The **cost** of a path is thought to be the **distance** between two nodes
- Originally used in ARPANET
- Now, it is being obsolete

Bellman-Ford Algorithm

- Each router maintains a routing table that contains
  - Preferred outgoing link to each destination in the subnet
  - The cost to that destination
- Each router is presumed to know the cost to all its **neighbors**
- The cost from a node x to a node x' can be determined by repeatedly asking for the cost from each neighbor in the path from x to x'

Bellman-Ford Algorithm (2)



- Determining the cost from u to z
  - u knows that c(u,v) = 2 and c(u,x) = 1
  - u asks v and x: What are c(v,z) and c(x,z), respectively
  - If c(v,z) = 8 and c(x,z) = 3,

    then the costs from u to z through v and x are

    c(u,v)+c(v,z) = 2+8 = 10 and
    c(u,x)+c(x,z) = 1+3 = 4, respectively

    So, u determines that c(u,z) = 4
  - But how do v and x determine c(v,z) and c(x,z)?
    - The same way as u did!

Bellman-Ford Algorithm in Action

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

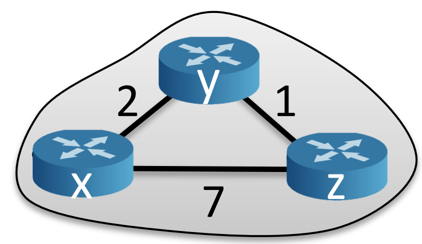
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

time →

Updating Distance Vector

- When a node detects link cost change
  - E.g., when a link is cut or an adjacent node is down
  - Distance vector in that node is updated
  - If the vector changes, the node notifies its neighbors

Link-State VS Distance Vector

Message complexity

- LS: with n nodes, E links, O(nE) messages sent
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^2)$ algorithm requires O(nE) messages
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect **link** cost
- each node computes only its **own** table

DV:

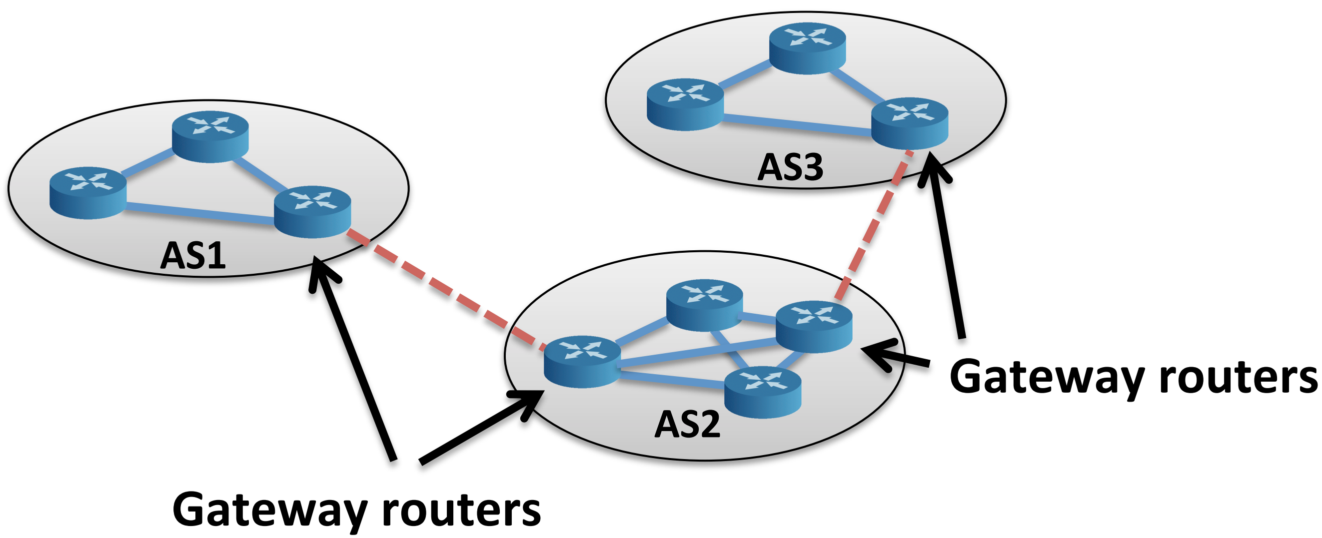
- DV node can advertise incorrect **path** cost
- each node's table used by others
  - error propagate through the network

Hierarchical Routing

- Some networks (such as the Internet) are too large, so the networks are divided into smaller **regions**
- Routers within the same region use the same routing protocols and algorithms
- All routers in the same region have routing information (e.g., link costs) of their own region
- Each region has at least one **gateway node**, which is used to communicate with other regions

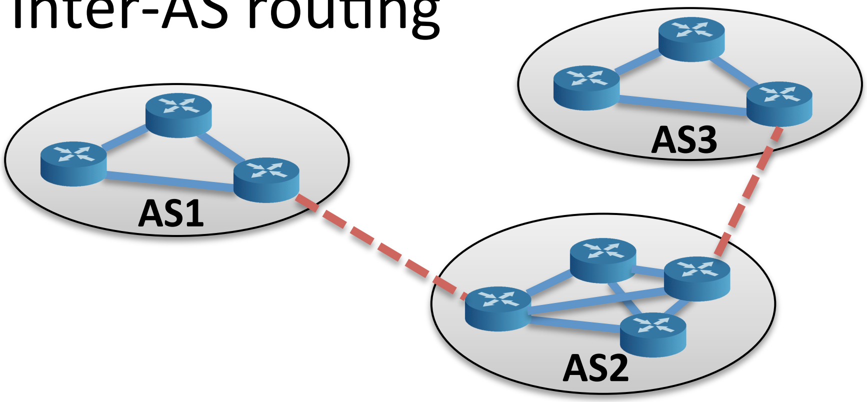
Hierarchical Routing (2)

- In computer network field, we call a region an **Autonomous System (AS)**
- Dividing the networks into ASes reduces the size of the routing table tremendously



Inter-Autonomous System Routing

- Suppose router in AS2 receives datagram for which destination is outside of AS2
  - Router should forward packet towards on of the gateway routers, but which one?
- AS2 has to learn which destinations are reachable through AS1 and which through AS3
- It also has to propagate this **reachability** info to all routers in AS2
- These are the tasks of Inter-AS routing
  - **BGP (RFC 1771)**
  - **Report!**



Routing Protocols

- **Interior Gateway Protocol (IGP)**: Routing within an AS (**Intra-AS routing**)
  - **Routing Information Protocol (RIP)**
    - Based on **Distance Vector** routing algorithm
    - Converge slowly
    - Widely used but getting obsolete
  - **Open Shortest Path First (OSPF)**
    - Based on **Link-State** and **Dijsktra's** algorithm
- **Exterior Gateway Protocol (EGP)**: Routing between different ASes (**Inter-AS routing**)
  - **Border Gateway Protocol (BGP)**: De facto standard in inter-AS routing

Internet Control Message Protocol

- In short: **ICMP**
- It is part of TCP/IP protocol suite
  - IP is responsible only for packet transfer
  - ICMP is used to send error and control messages among network devices
- ICMP for IPv6 is called **ICMPv6**
- Example messages
  - Destination Unreachable
  - Time Exceeded: Time-to-Live of a packet is exceeded
  - Echo Request / Reply: Request for an **echo**
  - Timestamp Request / Reply: Similar to echo request, but with timestamp of packet arrival time