# Distance Vector Routing Protocol

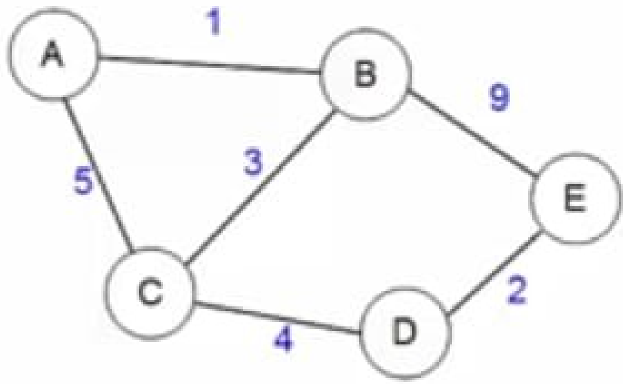
- Network Layer: Routing process
- Routing: Find the least cost path between two node
- Many approaches. Our focus: Dynamic, distributed algorithms
- Distance Vector Algorithm

# Background

- Also goes by the name Bellman-Ford algorithm
- Used in ARPAnet
- Later in Internet under the routing protocol standard RIP (Routing Information Protocol)
- Now, it is not used much

# Protocol Framework

- Initial state at a node: distance (cost) to neighbors is known
- Final state at a node: distance (cost) to all nodes is known, and also the next-hop
- Need to handle
  - What information to exchange? (message format)
  - How to act on a message?
  - When to send a message?



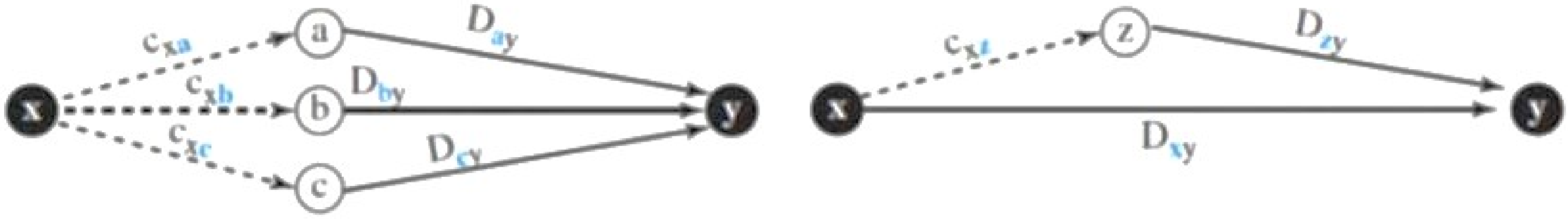
# Message Content

- Each node exchanges with all its neighbors "Routing Table" info
  - Destination and 'Estimated' cost to destination
  - Next hop information is not shared

# Action at a router

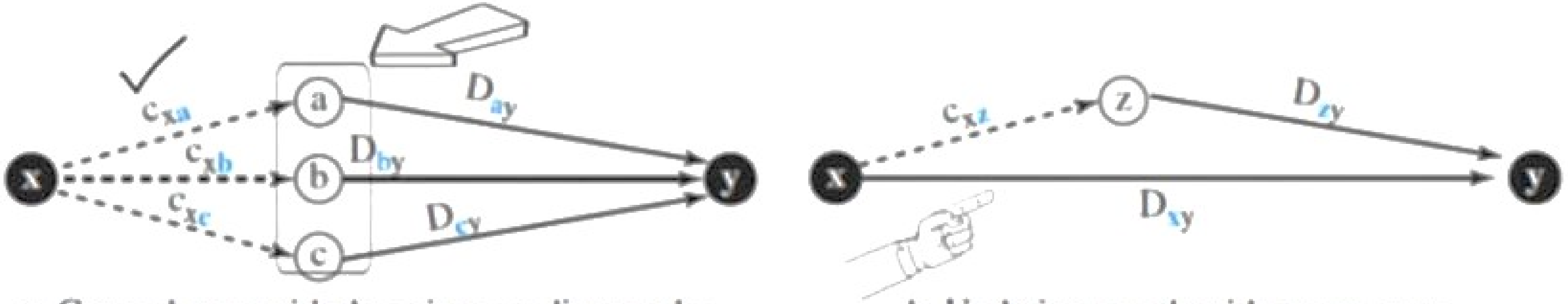
- On receiving a message from a neighbor v,
  - Update cost (estimate) to destinations based on above Bellman-ford equation; change next hop accordingly
  - For each y (destination in routing table of the received message)
    - $D_x(y) = \min\{\text{current estimate}, c(x,v) + D_v(y)\}$
  - Estimated costs finally converge to optimal cost after series of message exchanges

$$D_{xy} = \min\{D_{xy}, (c_{xz} + D_{zy})\}$$



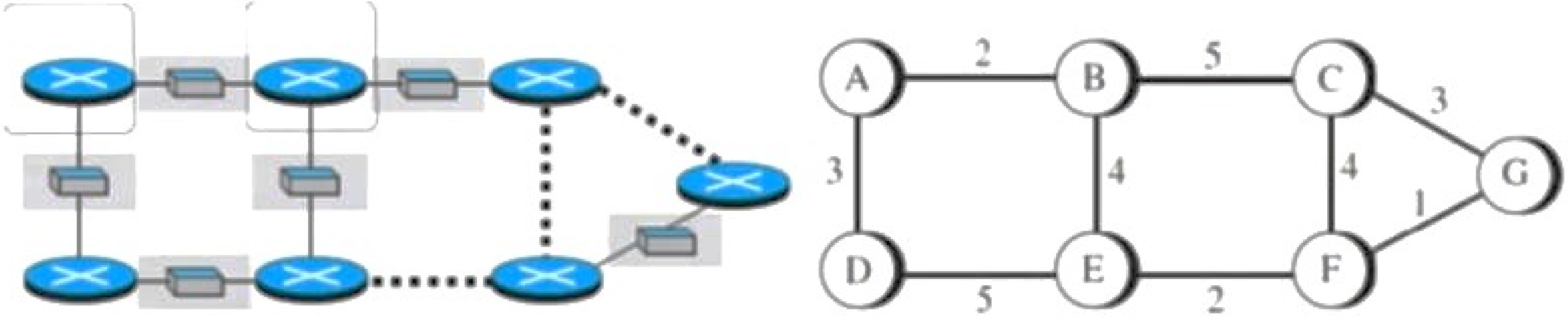
a. General case with three intermediate nodes

b. Updating a path with a new route



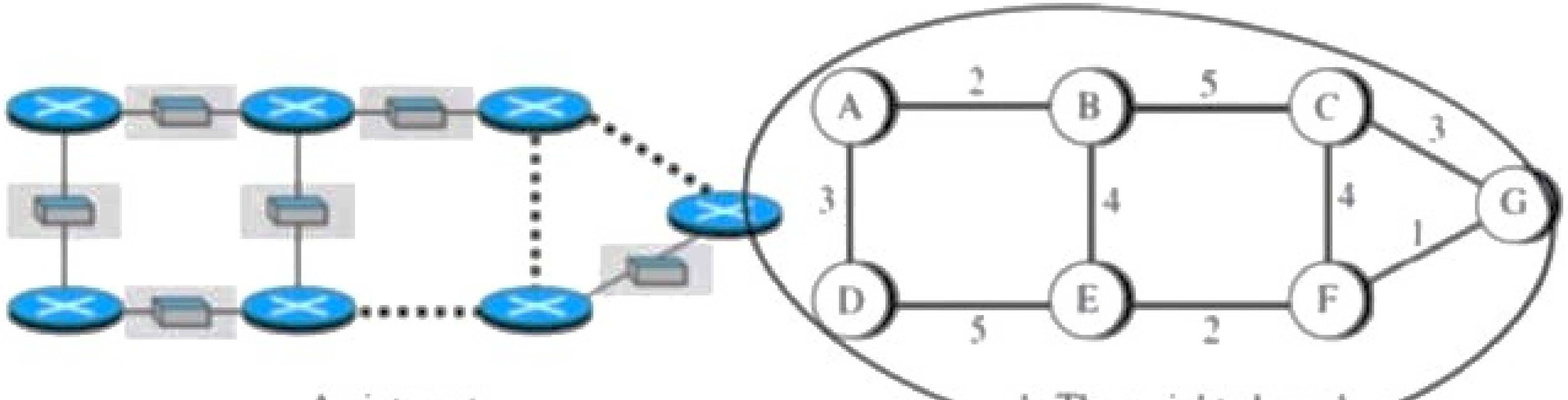

a. General case with three intermediate nodes

b. Updating a path with a new route



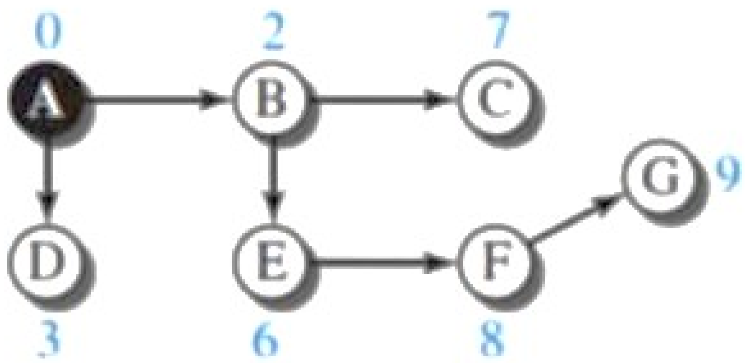

a. An internet

b. The weighted graph




a. An internet b. The weighted graph

## The distance vector corresponding to a tree

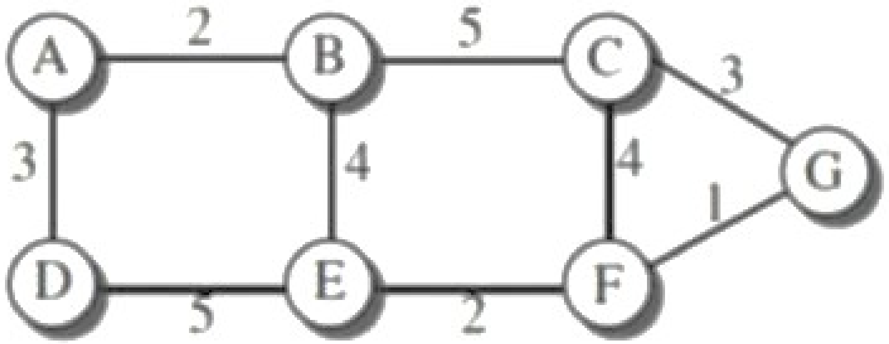



a. Tree for node A

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

b. Distance vector for node A

# Updating distance vectors




| | New B | Old B | A |
|---|---|---|---|
| A | 2 | 2 | 0 |
| B | 0 | 0 | 2 |
| C | 5 | 5 | ∞ |
| D | 5 | ∞ | 3 |
| E | 4 | 4 | ∞ |
| F | ∞ | ∞ | ∞ |
| G | ∞ | ∞ | ∞ |

$B[\ ] = \min(B[\ ],\ 2 + A[\ ])$

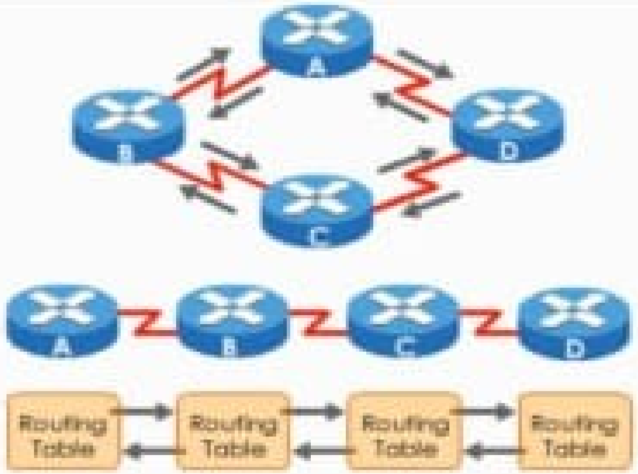
a. First event: B receives a copy of A's vector.

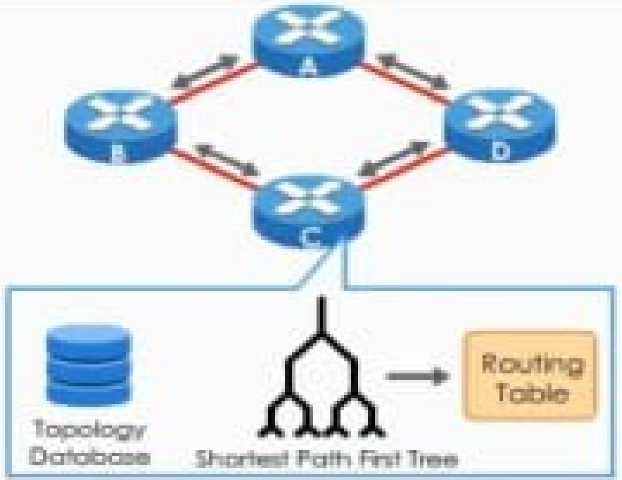
**Note**:
X[ ]: the whole vector

# Steps to follow

- Find the shortest path between source and the destination
- Exchanging vectors to find the nodes with minimal cost
- After updating the node it immediately send the update to all neighbor nodes.

# Distance-Vector vs Link-State Routing




| Distance-Vector(RIP, IGRP) | Link-State Routing(OSPF, IS-IS) |
|---|---|
| Each router only shares its routing table with its neighbors | Each router shares its complete map of the network with all other routers |
| Making routing decisions based on limited information | With a complete view of the network, routing decisions are more accurate and reliable |
| Easy to set up and configure | Difficult to set up and configure |
| Slow convergence | Fast convergence |
| The metric is typically based on the number of hops between the source and destination | The metric can be based on a variety of factors, such as the bandwidth or delay of a link |
| Suited for smaller networks | Suited for larger networks |