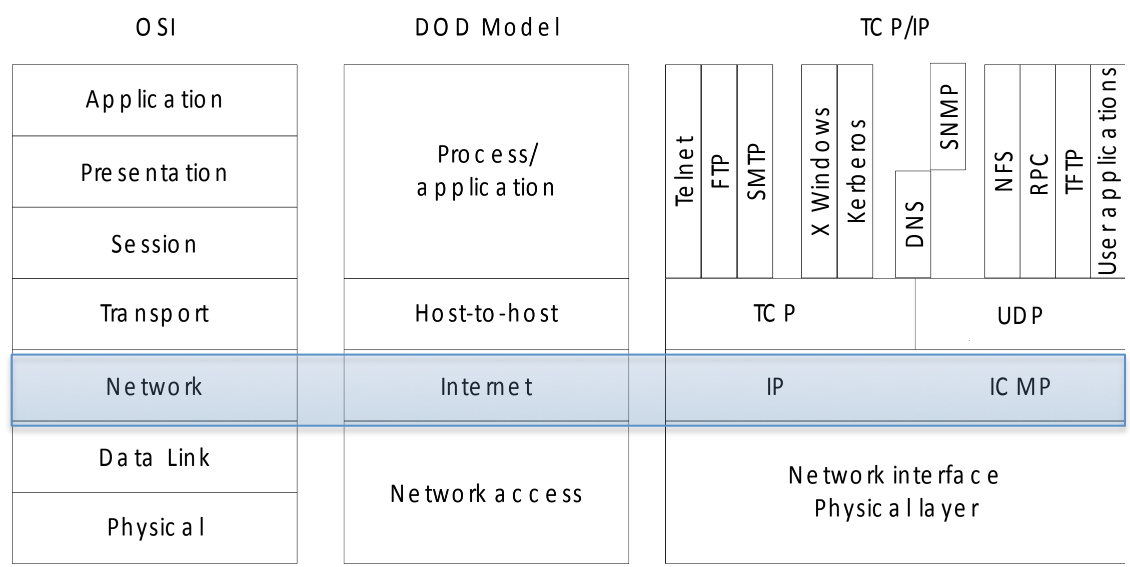
# Computer Network

Internetworking – Part 3

# Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed



# Routing

Feb 14 2017

## Forwarding versus Routing

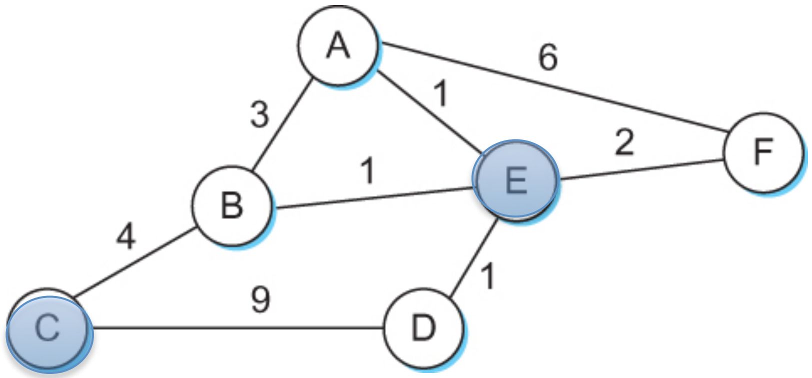
- Forwarding:
  - select an output port based on destination address and routing table
- Routing: more global
  - process by which forwarding tables are built (distributed algorithm)

- Routing means finding a suitable path for a packet from sender to destination(global and distributed)
- Forwarding is the process of sending the packet towards the destination based on routing information(local).

- Routing make sure tables are always up to date

# Routing

- Network as a Graph




- The basic problem of routing is to find the lowest-cost path between any two nodes.
  - Where the cost of a path equals the sum of the costs of all the edges that make up the path.

  - or the total number of hops of a path equals the sum of all the hops in that path

# Routing

- For a simple network, we can calculate all shortest paths and load them into some nonvolatile storage on each node.
- Such a static approach has several shortcomings
  - It does not deal with node or link failures
  - It does not consider the addition of new nodes or links
  - It implies that edge costs cannot change

- What is the solution?
  - Need a distributed and dynamic routing protocol
  - Routing protocols that dynamically discover network destinations.
  - Dynamic routing allows routing tables in routers to change if a router on the route goes down or if a new network is added.

# Dynamic Routing

- Routing Protocols running in routers continuously exchange network status updates between each other. done in 2 ways : Broadcast/Multicast (which one is more intelligent way?)
- Routing update messages are sent by the routing protocols, to continuously update the routing table whenever a network topology change happens.
- Examples of routing protocols are
  - Routing Information Protocol (RIP)
  - Enhanced Interior Gateway Routing Protocol (EIGRP) အိုက် ဂါ
  - Open Shortest Path First (OSPF)

# Dynamic Routing

- Routing protocols can be classified based on the algorithm they use
    - Distance Vector (RIP, RIPv2, EIGRP)
    - Link State (OSPF)

, ISIS = Intermediate System to Intermediate System

- They can also be classified based on whether they can route between multiple autonomous systems or just within one autonomous system (one domain, one admin, one ISP, etc.)
    - Interior Gateway Protocols (IGP)
        - RIP, EIGRP, OSPF
    - Exterior Gateway Protocol (EGT)
        - BGT (Border Gate Protocol)

# Dynamic Routing
# Classified based on the Algorithm

just neighbor

everybody not just neighbor

| **Distance Vector** | **Link State** |
|---|---|
| RIP, RIPv2, IGRP, EIGRP | OSPF, ISIS |
| Routers communicate with neighbor routers advertising networks as measures of distance and vector | Routers communicate with all other routers exchanging link-state information to build a topology or the entire network |
| Distance = Metric<br>Vector = Direction (Interface) | Link-state = interface connections or "links" to other routers and networks |
| Best for:<br>- simple, flat design, non-hierarchical networks<br>- minimum administrator knowledge<br>- convergence time is not an issue | Best for:<br>- large, hierarchical networks<br>- advanced administrator knowledge<br>- convergence time is crucial |
| Knowledge of the network from directly connected neighbors | Routers have a complete view of the network, knowledge of the entire topology |
| Send periodic updates of entire routing table | Send triggered partial updates |

conv.time is slower

c.time is faster

# Dynamic Routing
# Intra or Inter Autonomous System Routing

**Routing Protocols**

The protocols of routers,
Routers Talking to Routers on the Network
Sharing Routing Information With Each Other

**IGP - Interior Gateway Protocols**

**Intra =>** RIP RIPv2 IGRP EIGRP OSPF ISIS

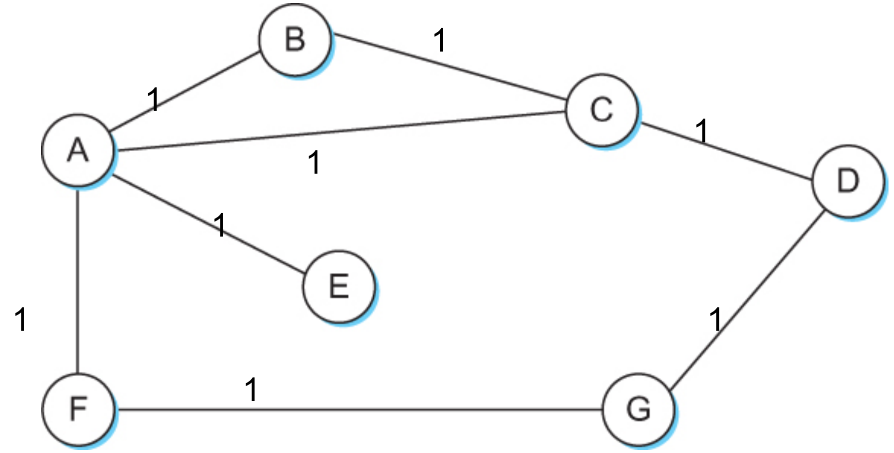
**EGP - Exterior Gateway Protocols**

**Inter =>** BGP

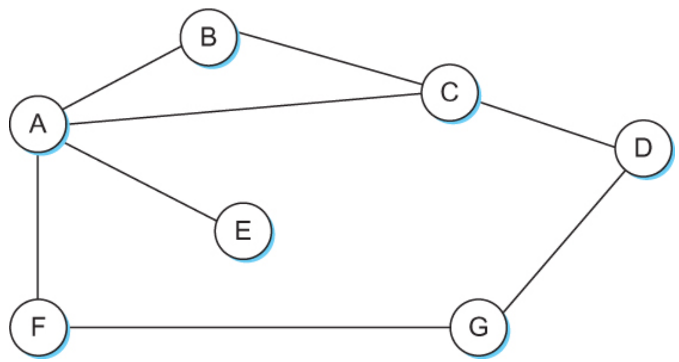
# Distance Vector

- Each node constructs a one dimensional array (a vector, or an entry in a table) containing the "distances" (costs) to all directly connected nodes and distributes that vector to its immediate neighbors
- Starting assumption is that each node knows the cost of the link to each of its directly connected neighbors (all will be 1 in this example)

from A to (each inside) A(B,C,E,F) = A(1,1,1,1)



# Distance Vector

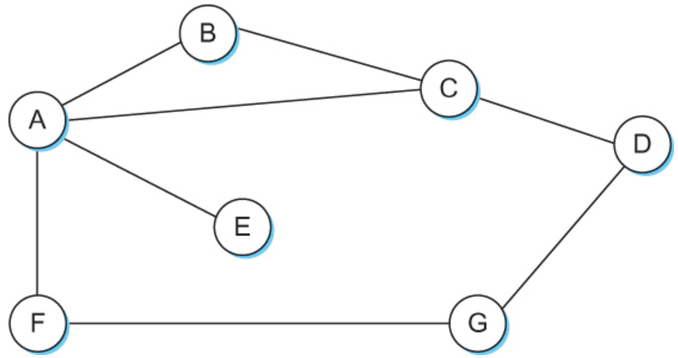



A knows nothing about D or G because no neighbor and don't know yet.

| Destination | Cost | NextHop |
| --- | --- | --- |
| B | 1 | B |
| C | 1 | C |
| D | $\boxed{\infty}$ | — |
| E | 1 | E |
| F | 1 | F |
| G | $\boxed{\infty}$ | — |

Initial routing table at node A

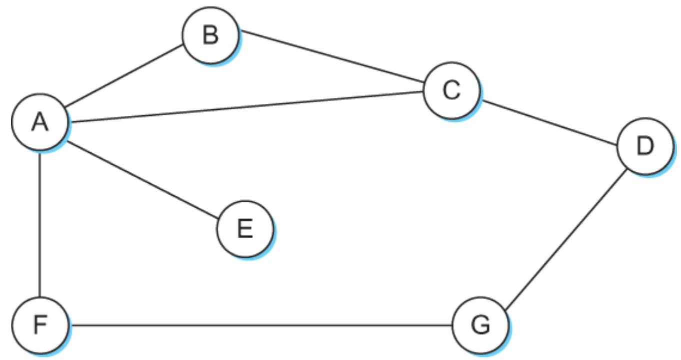
# Distance Vector




| Information Stored at Node | Distance to Reach Node | | | | | | |
|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G |
| initial distances stored at node A — A | 0 | 1 | 1 | ∞ | 1 | 1 | ∞ |
| B | 1 | 0 | 1 | ∞ | ∞ | ∞ | ∞ |
| initial distances stored at node C — C | 1 | 1 | 0 | 1 | ∞ | ∞ | ∞ |
| D | ∞ | ∞ | 1 | 0 | ∞ | ∞ | 1 |
| E | 1 | ∞ | ∞ | ∞ | 0 | ∞ | ∞ |
| F | 1 | ∞ | ∞ | ∞ | ∞ | 0 | 1 |
| G | ∞ | ∞ | ∞ | 1 | ∞ | 1 | 0 |

Initial distances stored at each node (global view)

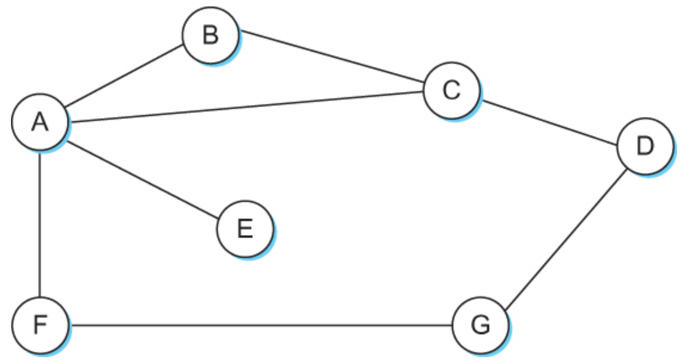
# Distance Vector




| Destination | Cost | NextHop |
|---|---|---|
| B | 1 | B |
| C | 1 | C |
| D | 2 | C |
| E | 1 | E |
| F | 1 | F |
| G | 2 | F |

Final routing table at node A

# Distance Vector




by now A has realized that it can get to D via C in 2 hop

| Information Stored at Node | Distance to Reach Node | | | | | | |
|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G |
| A | 0 | 1 | 1 | $\boxed{2}$ | 1 | 1 | 2 |
| B | 1 | 0 | 1 | 2 | 2 | 2 | 3 |
| C | 1 | 1 | 0 | 1 | 2 | 2 | 2 |
| D | 2 | 2 | 1 | 0 | 3 | 2 | 1 |
| E | 1 | 2 | 2 | 3 | 0 | 2 | 3 |
| F | 1 | 2 | 2 | 2 | 2 | 0 | 1 |
| G | 2 | 3 | 2 | 1 | 3 | 1 | 0 |

Final distances stored at each node (global view)

# Distance Vector

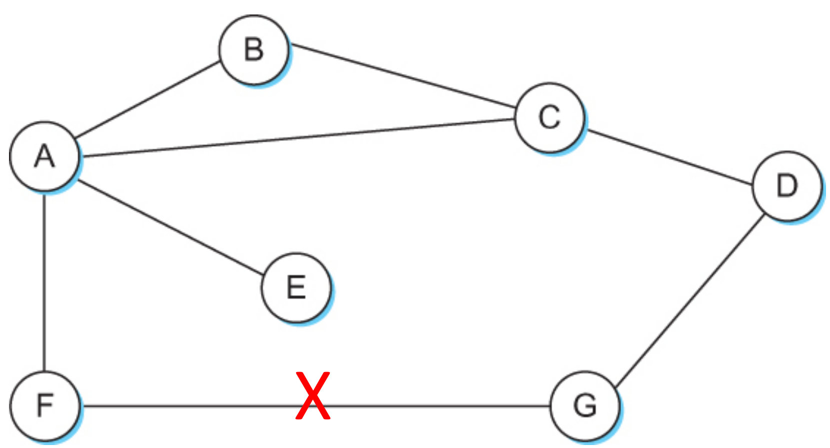
- This distance vector routing algorithm is based on Bellman-Ford algorithm.
- Every T seconds (or minutes) each router sends its table to its neighbor, and each router then updates its table based on the new information
- Problems include fast response to good news and slow response to bad news. Also too many messages to update.
- "Convergence time is the time needed for all routers within a single routing domain to receive, process and build their routing table" (Osterloh, 2002).

**Convergence time def: if A has something new to say, it will be after xx second, changing table with new updates (this updating time is longer than "link state" based routing protocol - which process more rapidly)**

- The convergence time of routing protocols based on DV (or Bellman-Ford) algorithm is slower than the Link-State based routing protocols.

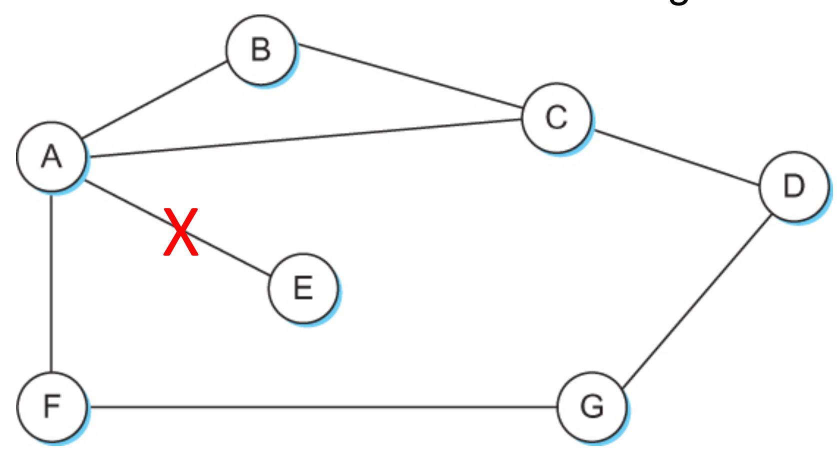
# Distance Vector

- When a node detects a link failure
  - F detects that link to G has failed
  - F sets distance to G to infinity and sends update to A
  - A sets distance to G to infinity since it uses F to reach G
  - A receives periodic update from C with 2-hop path to G
  - A sets distance to G to 3 and sends update to F
  - F decides it can reach G in 4 hops via A
  - At this point the network is said to be converged



# Distance Vector

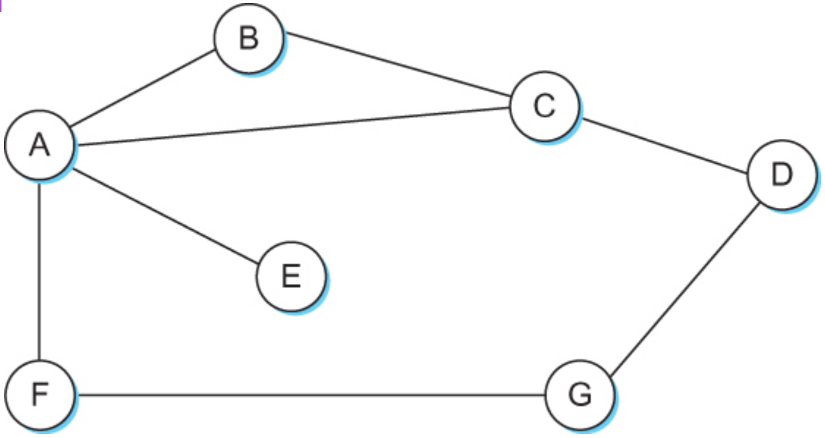
- Slightly different circumstances can prevent the network from stabilizing (converging)
  - Suppose the link from A to E goes down
  - In the next round of updates, A advertises a distance of infinity to E, but B and C advertise a distance of 2 to E (both B & C got this information initially from A)
  - Depending on the exact timing of events, the following might happen
    - Node B, upon hearing that E can be reached in 2 hops from C, concludes that it can reach E in 3 hops and advertises this to A
    - Node A concludes that it can reach E in 4 hops and advertises this to C
    - Node C concludes that it can reach E in 5 hops; and so on.
    - This cycle stops only when the distances reach some number that is large enough to be considered infinite
      - **Count-to-infinity problem**
      - **Or also called routing loops**



# Count-to-infinity Problem

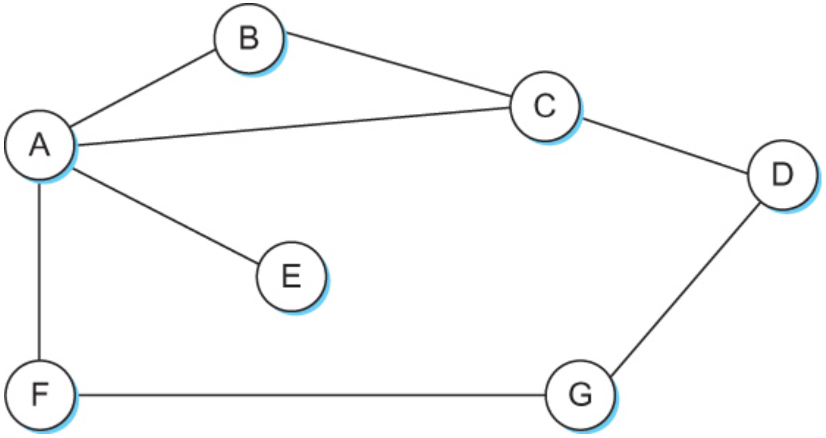
1. Use some relatively small number as an approximation of infinity
   – For example, the maximum number of hops to get across a certain network is never going to be more than 16

2. One technique to improve the time to stabilize routing is called *split horizon*
   – When a node sends a routing update to its neighbors, it does not send those routes it learned from each neighbor back to that neighbor
   – For example, if B has the route (E, 2, A) in its table, then it knows it must have learned this route from A, and so whenever B sends a routing update to A, it does not include the route (E, 2) in that update

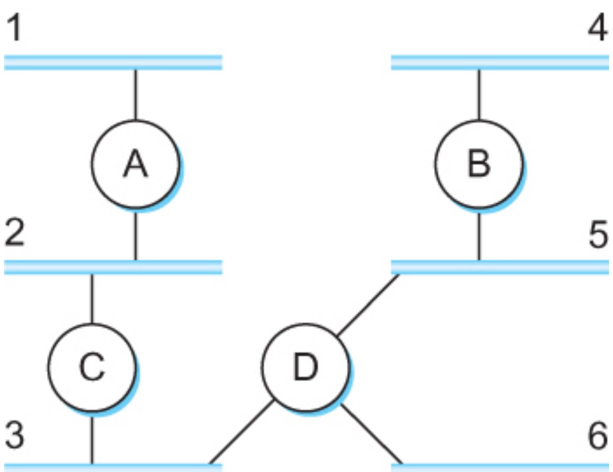


# Count-to-infinity Problem

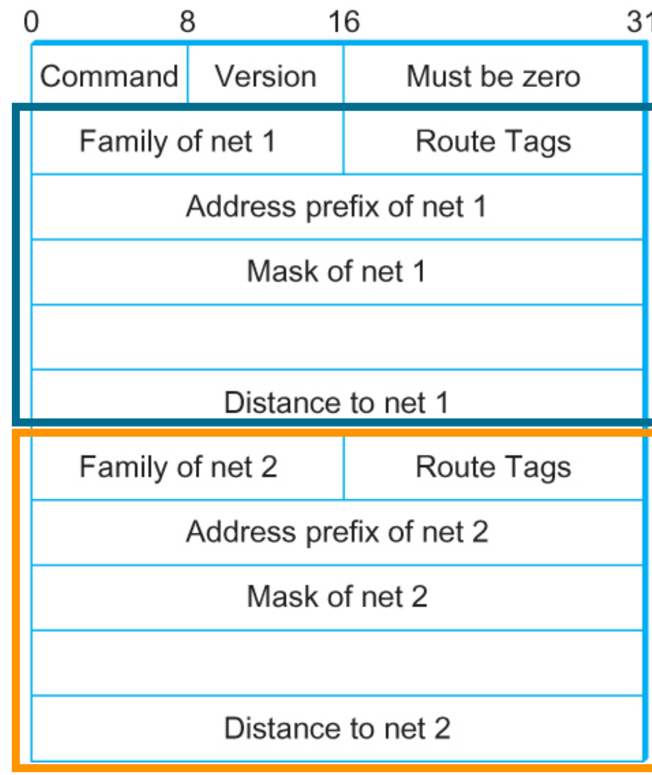
- In a stronger version of split horizon, called *split horizon with poison reverse*
  - B actually sends that back route to A, but it puts negative information in the route to ensure that A will not eventually use B to get to E
  - For example, B sends the route (E, ∞) to A



# Routing Information Protocol (RIP)



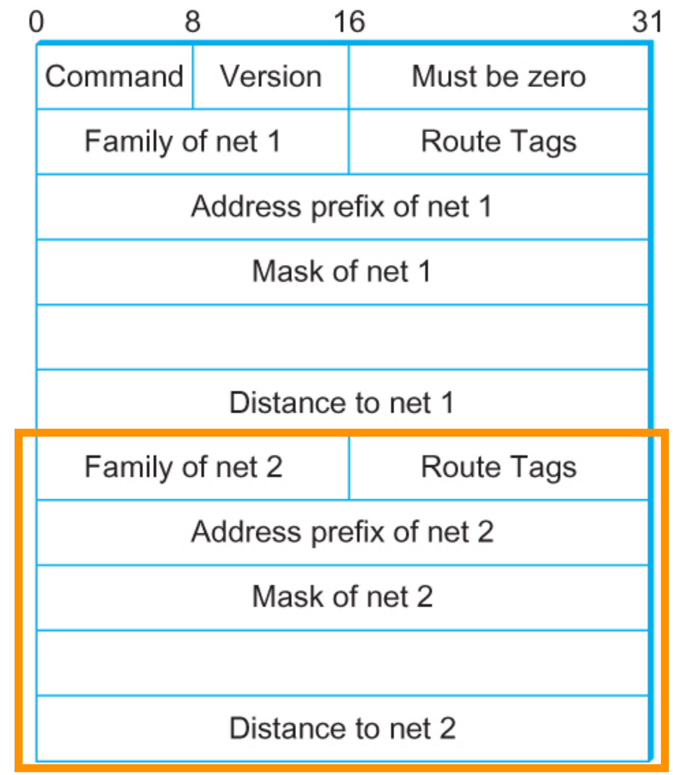

Example Network running RIP




RIPv2 Packet Format

# Routing Information Protocol (RIP)

- **Command** - Indicates whether the packet is a request or a response.
- **Version number** - Specifies the RIP version used.
- **Zero** - This field is not actually used.
- **Family Address identifier**- Specifies the family used. The AFI for IP is 2.
- **Address** - Specifies the IP address for the entry (network).
- **Subnet Mask** – Specifies the subnet mask of the network
- **Metric** (distance to net 1) - Indicates how many internetwork hops (routers) have been traversed in the trip to the destination.




RIPv2 Packet Format

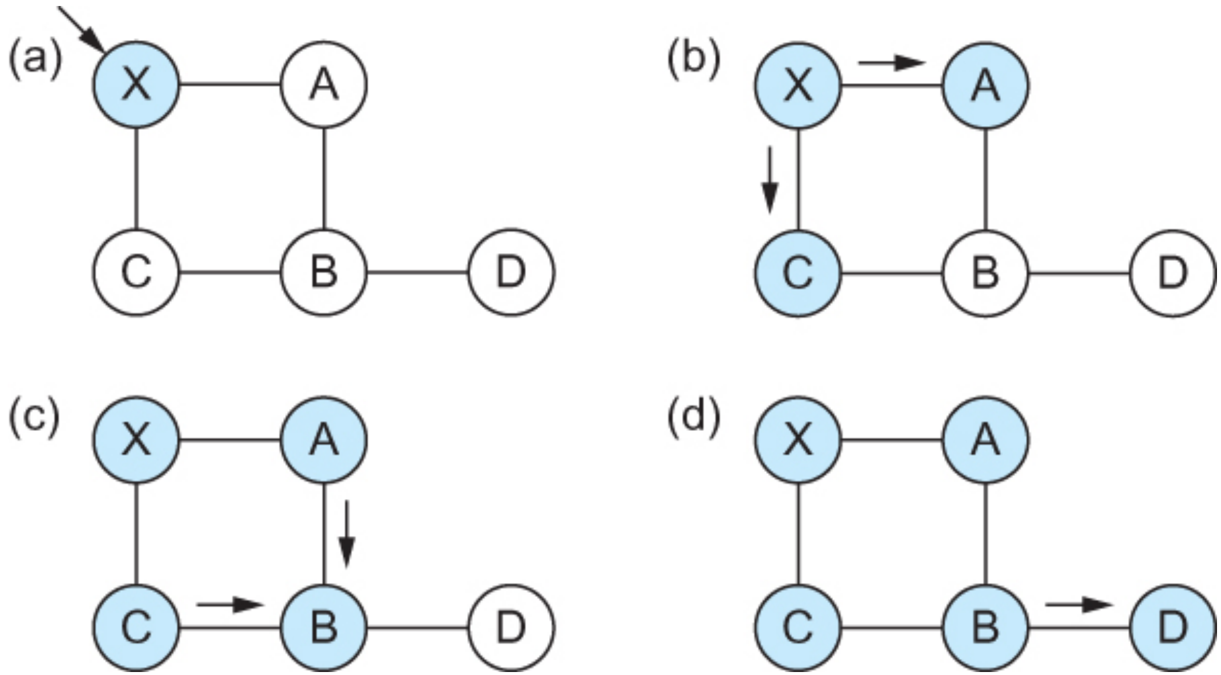
# Link State Routing

Strategy: Send to all nodes (not just neighbors) information/updates about directly connected links (not the entire routing table).

- Link State Packet (LSP), aka Update Packets
  1. id of the node that created the LSP
  2. a list of directly connected neighbors of that node along with cost of link to each directly connected neighbor
  3. sequence number (SEQNO)
     - a larger sequence number implies a more recent one
  4. time-to-live (TTL) for this packet
     - to ensure that old link-state information is eventually removed from the network
- These packets implement the flooding of link state advertisements.

# Link State Routing

Reliable Flooding




Flooding of link-state packets:
(a) LSP arrives at node X;
(b) X floods LSP to A and C;
(c) A and C flood LSP to B (but not X);
(d) Flooding is complete

# Link State Routing

- Reliable Flooding Rules
  - Store most recent LSP from each node
  - Forward LSP to all nodes except to the one that sent it
  - Generate new LSP periodically; increment SEQNO
  - Start SEQNO at 0 when reboot
  - Decrement TTL of each stored LSP; discard when TTL=0

**2 or 3 slides are missing with figures where (each router use link state database adn runs Dijkstra shortest path first algorithm … and more )**

# Open Shortest Path Routing

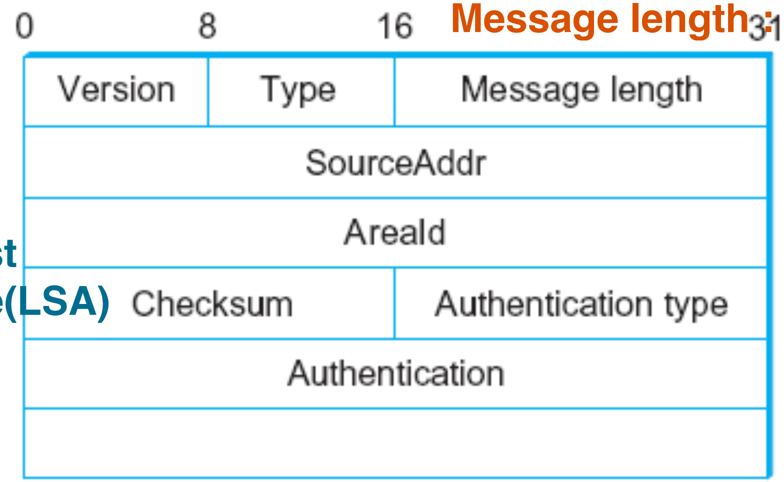
- An extension of the Short Path Routing (SPF) which is another name for Link-State Routing
  - Link-State Routing is based on Dijkstra's Algorithm
  - "Open" refers to the fact that it is an open, non proprietary standard created by IETF
- Adds more feature to the basic Link-State Routing
  - Authenticates the routing messages
    - Entire network can be impacted by a bad message from one node
    - Uses strong cryptography authentication
  - Provides additional hierarchy by allowing a domain to be partitioned into areas
    - Routers only need to know about the nodes in a certain area
  - Uses load balancing to distribute traffic evenly over routers

# Open Shortest Path First (OSPF)

**- RIP and OSPF are examples of intradomain protocols**

**Message length: including the header**

**Type=1 -> HELLO**
**Type =2 -> DB description**
**Type=3 -> link state request**
**Type=4 -> link state update(LSA)**
**Type=5 -> Link state ACK**




OSPF Header Format

# Open Shortest Path First (OSPF)

details will be on final

Type 1 LSA advertises the cost of links between routers.

LS Age is like the TTL except that it counts up.

| | | | | |
|---|---|---|---|---|
| LS Age | | | Options | Type=1 |
| Link-state ID | | | | |
| Advertising router | | | | |
| LS sequence number | | | | |
| LS checksum | | | Length | |
| 0 | Flags | 0 | Number of links | |
| Link ID | | | | |
| Link data | | | | |
| Link type | | Num_TOS | Metric | |
| Optional TOS information | | | | |
| More links | | | | |

length is the length in bytes of the complete LSA.

LS checksum is used to make sure the data is not corrupted. it covers all fields in the packets except LS Age

OSPF Link State Advertisement (LSA)

# Open Shortest Path First (OSPF)

<table>
<tr><td colspan="4">LS Age</td><td colspan="2">Options</td><td colspan="2">Type=1</td></tr>
<tr><td colspan="8">Link-state ID</td></tr>
<tr><td colspan="8">Advertising router</td></tr>
<tr><td colspan="8">LS sequence number</td></tr>
<tr><td colspan="4">LS checksum</td><td colspan="4">Length</td></tr>
<tr><td>0</td><td>Flags</td><td colspan="2">0</td><td colspan="4">Number of links</td></tr>
<tr><td colspan="8">Link ID</td></tr>
<tr><td colspan="8">Link data</td></tr>
<tr><td colspan="2">Link type</td><td colspan="2">Num_TOS</td><td colspan="4">Metric</td></tr>
<tr><td colspan="8">Optional TOS information</td></tr>
<tr><td colspan="8">More links</td></tr>
</table>

OSPF Link State Advertisement (LSA)

# Summary

- We have looked at some of the issues involved in building scalable and heterogeneous networks by using switches and routers to interconnect links and networks.
- To deal with heterogeneous networks, we have discussed in details the service model of Internetworking Protocol (IP) which forms the basis of today's routers.
- We have discussed in details two major classes of routing algorithms
  - Distance Vector
  - Link State
- Hubs, Switches and Routers Explained

# Administrative Distance Number

**Administrative distance (AD)** is an arbitrary numerical value assigned to a routing protocol, a static route or a directly-connected route based on its perceived quality of routing. The administrative distance (AD) value is often used by Cisco routers to **determine the "best" route that should be used when multiple paths to the same destination exist**. A routing protocol with a lower administrative distance is considered "better" and is given priority over routing protocols with higher administrative distances.

| Routing Protocol | Administrative distance |
|---|---|
| Directly connected interface | 0 |
| Static route out an interface | 1 |
| Static route to next-hop address | 1 |
| DMNR - Dynamic Mobile Network Routing | 3 |
| EIGRP summary route | 5 |
| External BGP | 20 |
| Internal EIGRP | 90 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| Routing Information Protocol (RIP) | 120 |
| Exterior Gateway Protocol (EGP) | 140 |
| On Demand Routing (ODR) | 160 |
| External EIGRP | 170 |
| Internal BGP | 200 |
| Floating Static Route (ex. DHCP-learned) | 254 |
| Unknown | 255 |