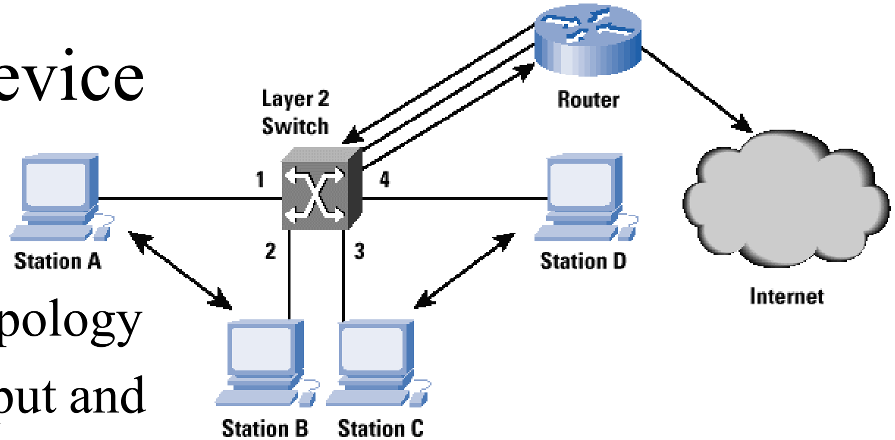
Internetworking

(Bridging, Routing, fragmentation, segmentation)

Chun-Jen (James) Chung

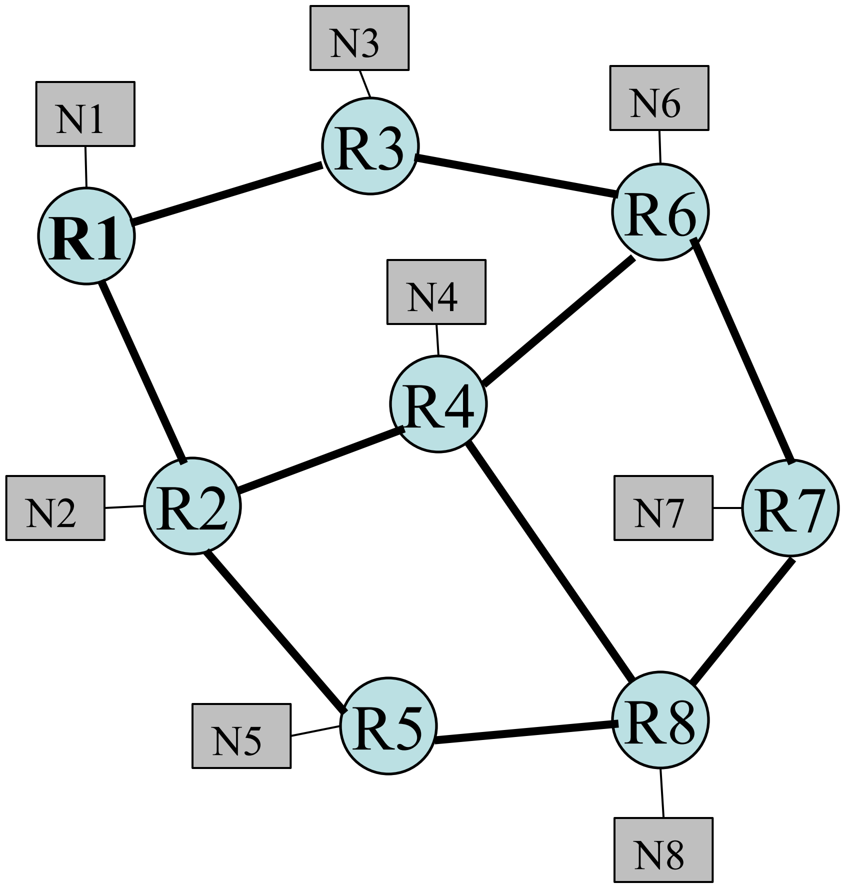
Arizona State University

Bridging (switching) vs. Routing



- Switches (Bridges) – layer 2 device
  - Manage traffic between LANs
  - Know only about MAC addresses
  - Have no knowledge of larger network topology
  - **Forwarding**: taking a packet from an input and sending it out on the appropriate output *port*
- Routers – layer 3 device
  - Manage traffic between networks
  - Know about global space of network addresses
  - Can make better decision based on the knowledge of the network topology
  - **Routing**: building up the *tables* to determine the correct output for a packet

Next Hop Routing within Networks



R1's routing table

Network	NextHop
1	–
2	2
3	3
4	2
5	2
…	

R2's routing table

Network	NextHop
1	1
2	–
3	1
4	4
5	5
…	

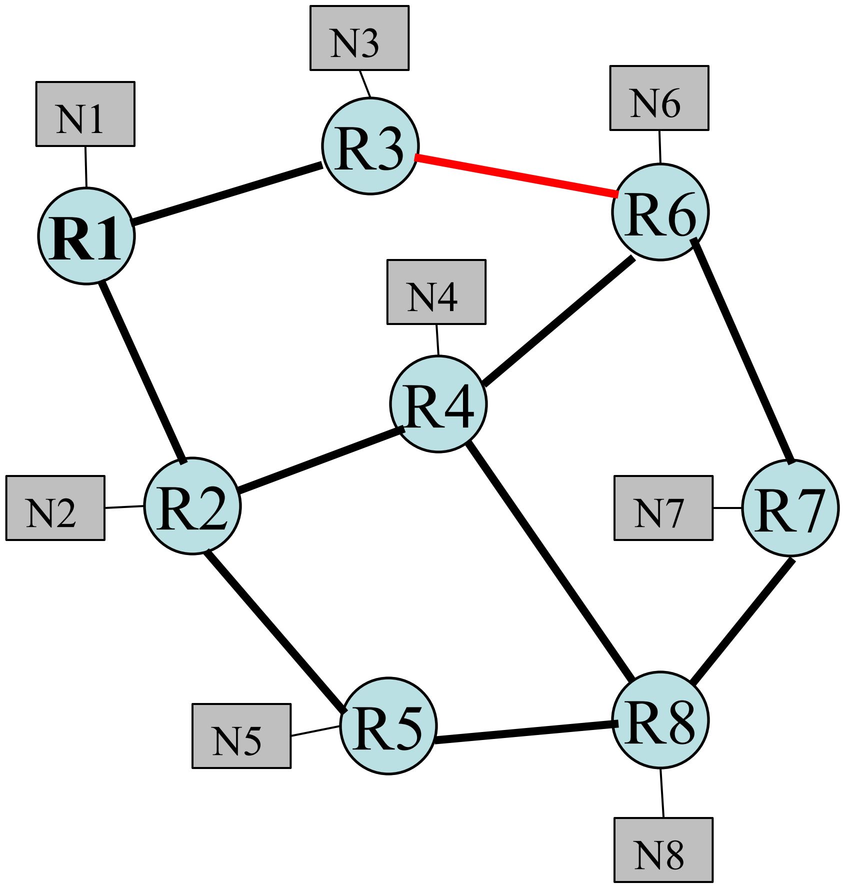
R3's routing table

Network	NextHop
1	1
2	1
3	–
4	6
5	1
…	

R4's routing table

Network	NextHop
1	2
2	2
3	6
4	–
5	2
…	

Modified Routing



R1's routing table

Network	NextHop
1	–
2	2
3	3
4	2
5	2
…	

R2's routing table

Network	NextHop
1	1
2	–
3	1
4	4
5	5
…	

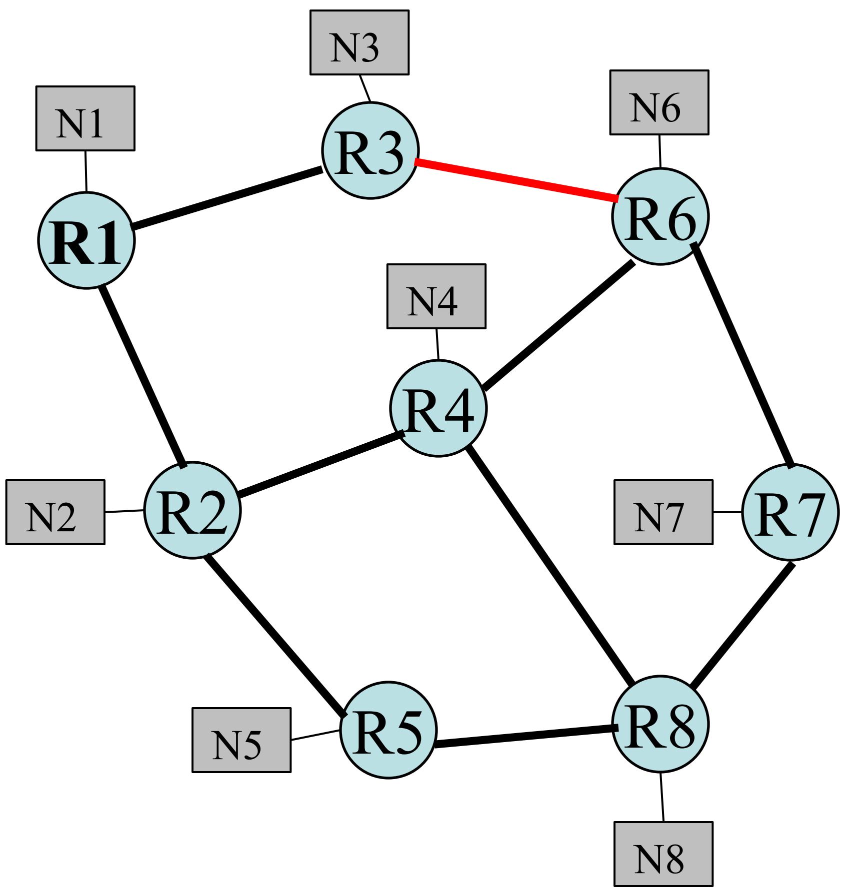
R3's routing table

Network	NextHop
1	1
2	1
3	–
4	**1**
5	1
…	

R4's routing table

Network	NextHop
1	2
2	2
3	**2**
4	–
5	2
…	

Default Routing within Networks



R1's routing table

Network	NextHop
1	–
2	2
3	3
*	2
…	

R2's routing table

Network	NextHop
1	1
2	–
3	1
4	4
5	5
…	

R3's routing table

Network	NextHop
1	1
2	1
3	–
4	**1**
5	1
…	

R4's routing table

Network	NextHop
1	2
2	2
3	**2**
4	–
5	2
…	

Routing

- Static Routing
  - Manually-configured routing entry
  - Used when routing choices are limited or unchanging
  - Often used in **small networks**
  - No extra resources needed
- Dynamic Routing
  - Automatically adapts to network topology changes
  - Suitable for simple and **complex topologies**
  - Require extra resources (CPU, memory, bandwidth)
  - Used in both **Interior Gateways** and **Exterior Gateways**

Static Routing

- route command

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         172.24.55.193   0.0.0.0         UG    0      0        0 eth0
link-local      *               255.255.0.0     U     1000   0        0 eth0
172.24.55.0     *               255.255.255.240 U     1      0        0 eth1
172.24.55.128   *               255.255.255.240 U     1      0        0 eth2
172.24.55.134   172.24.55.193   255.255.255.255 UGH   0      0        0 eth0
172.24.55.192   *               255.255.255.240 U     1      0        0 eth0
```

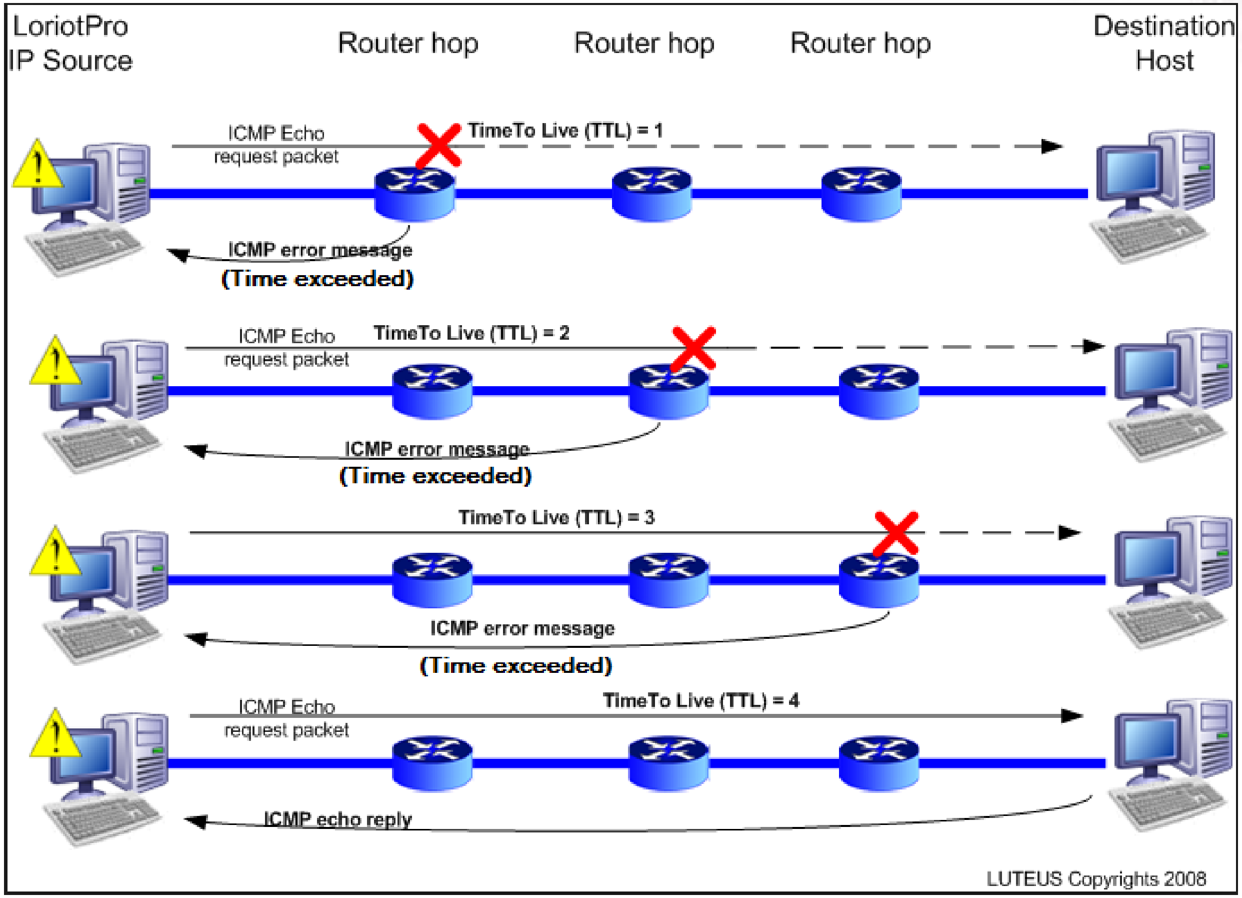
- Flags
  - U: the route is up
  - G: the route is to a gateway/rotuer
  - H: the route is to a host
- Metric: the number of hops to the destination.
- Ref: the number of TCP connections currently using this route.
- Use: the number of packets that have passed via this route.

traceroute

- A diagnostic tool for displaying the **route (path)** and measuring transit delays of packets across an IP network.

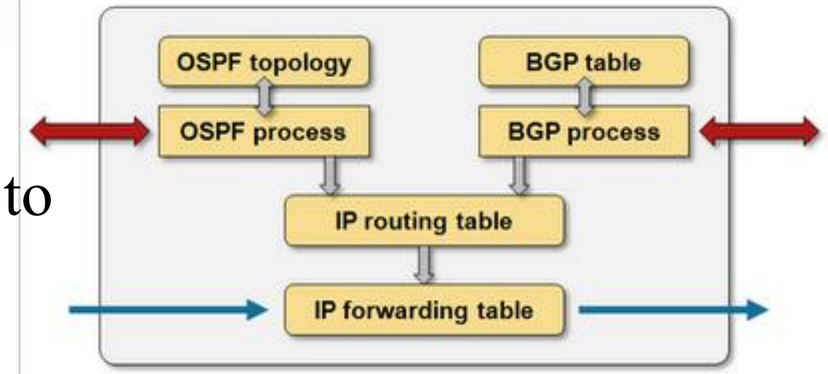
```
ubuntu@VM-Client:~$ traceroute www.google.com
traceroute to www.google.com (74.125.224.180), 30 hops max, 60 byte packets
 1  VM-GW.local (172.24.55.4)  2.893 ms  2.827 ms  2.739 ms
 2  172.24.55.193 (172.24.55.193)  4.852 ms  4.802 ms  4.721 ms
 3  10.211.16.1 (10.211.16.1)  10.408 ms  10.332 ms  10.341 ms
 4  172.19.107.98 (172.19.107.98)  10.000 ms  9.972 ms  9.932 ms
 5  172.19.107.77 (172.19.107.77)  9.916 ms  9.934 ms  9.903 ms
 6  172.19.107.74 (172.19.107.74)  9.796 ms  5.317 ms  5.483 ms
 7  172.30.251.1 (172.30.251.1)  7.427 ms  3.862 ms  3.802 ms
 8  172.19.100.169 (172.19.100.169)  6.392 ms  8.992 ms  8.954 ms
 9  172.30.101.2 (172.30.101.2)  8.928 ms  8.908 ms  8.871 ms
10  149.169.2.37 (149.169.2.37)  9.428 ms  9.439 ms  9.419 ms
11  198.71.47.17 (198.71.47.17)  20.068 ms  20.074 ms  20.069 ms
12  * * *
13  209.85.248.185 (209.85.248.185)  30.582 ms  18.688 ms  23.020 ms
14  72.14.236.11 (72.14.236.11)  22.929 ms  22.908 ms  22.903 ms
15  lax02s01-in-f20.1e100.net (74.125.224.180)  41.628 ms  41.645 ms  41.663 ms
```

traceroute – how it works?



Routing table vs. Forwarding table

- Routing table
  - A data table that lists the routes to particular network destination.
  - Contains information about the topology of the network around it.
- Forwarding table
  - Contains only the routes which are chosen by the routing algorithm as preferred routes for packet forwarding.



Routing Table

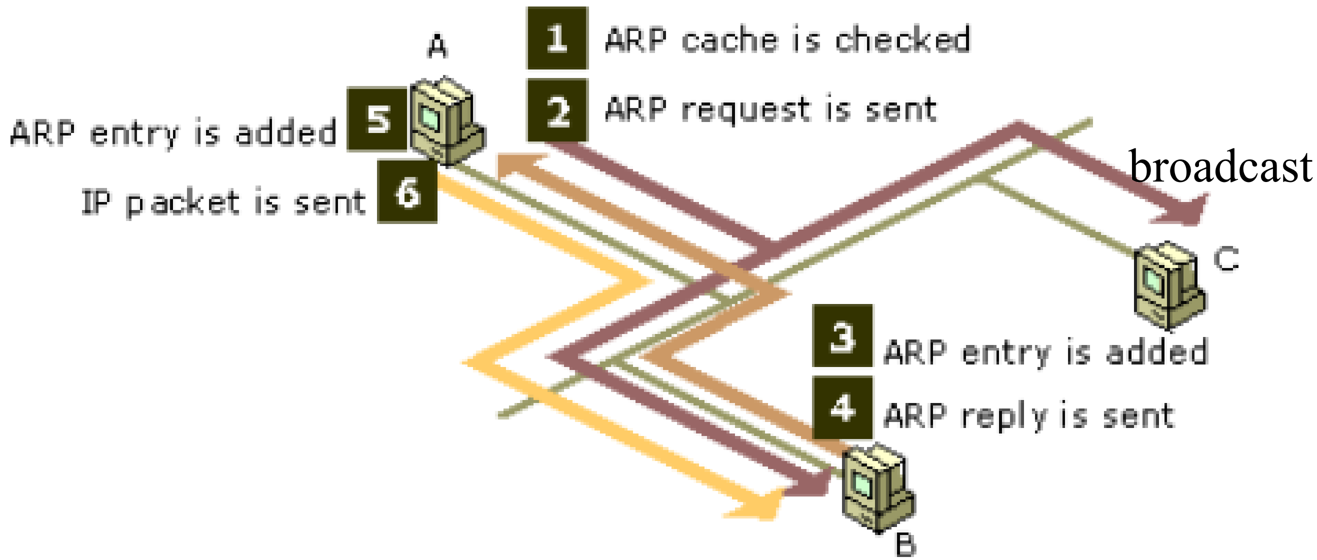
Network	Next Hop
10	172.16.245.10

Forwarding Table

Network	iface	MAC Address
10	eh1	8:0:2b:e3:b:1:2

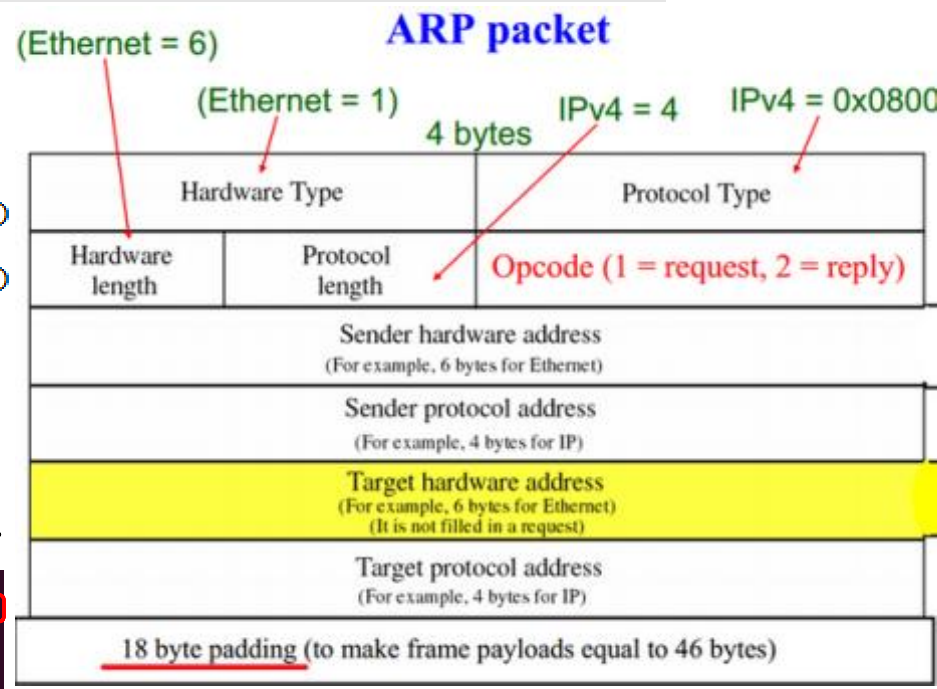
ARP

- Address Resolution Protocol
  - Resolving IPv4 addresses to MAC addresses
  - Maintaining a cache of mappings



ARP Packet

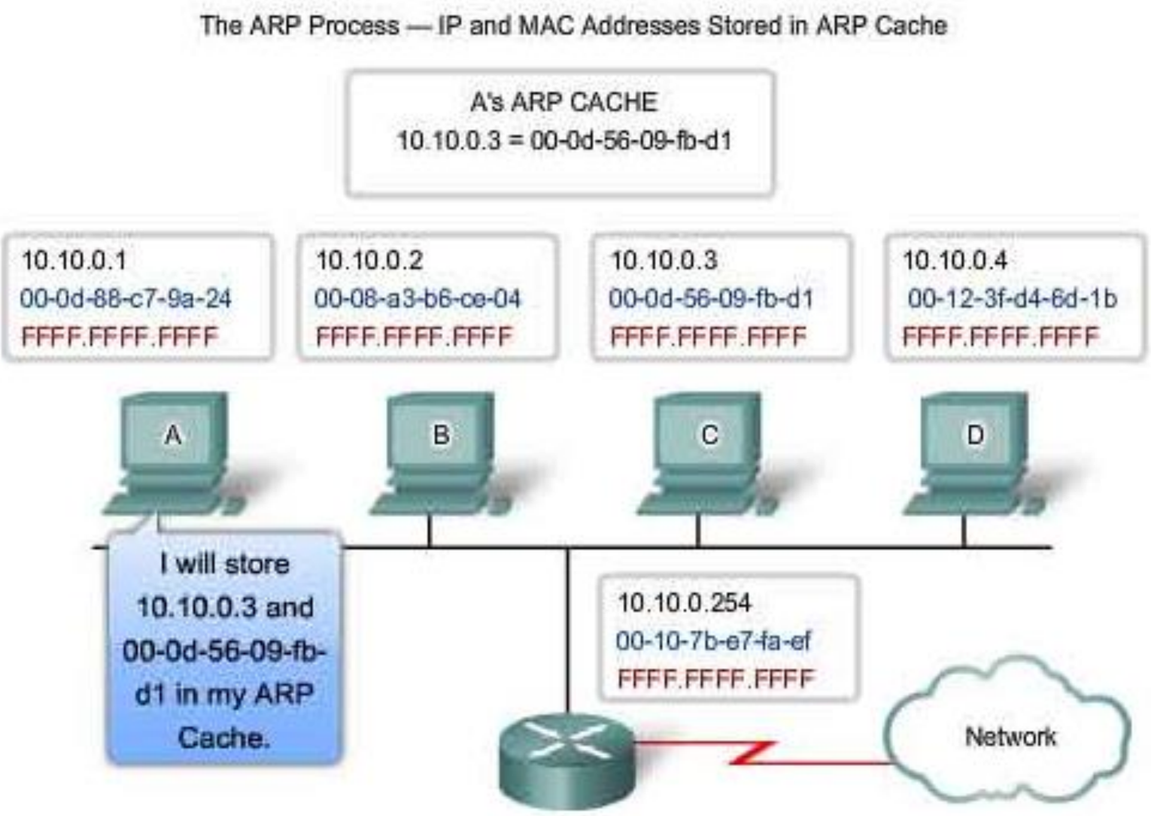
```
⊞ Frame 74: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
⊞ Ethernet II, Src: 00:26:c7:5c:45:7c (00:26:c7:5c:45:7c), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: False]
    Sender MAC address: 00:26:c7:5c:45:7c (00:26:c7:5c:45:7c)
    Sender IP address: 192.168.1.10 (192.168.1.10)
    Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1 (192.168.1.1)
```



ARP cache (arp table): removed after certain time.

```
root@VM-GW:~# arp -a
VM-Server.local (172.24.55.134) at fa:16:3e:04:f4:56 [ether] on eth2
? (172.24.55.194) at fa:16:3e:26:e8:7e [ether] on eth0
? (172.24.55.130) at fa:16:3e:6a:a8:09 [ether] on eth2
? (172.24.55.2) at fa:16:3e:13:20:8f [ether] on eth1
? (172.24.55.193) at fa:16:3e:0a:54:c1 [ether] on eth0
```

ARP Process in a LAN



Courtesy of http://www.highteck.net/EN/Ethernet/Ethernet.html

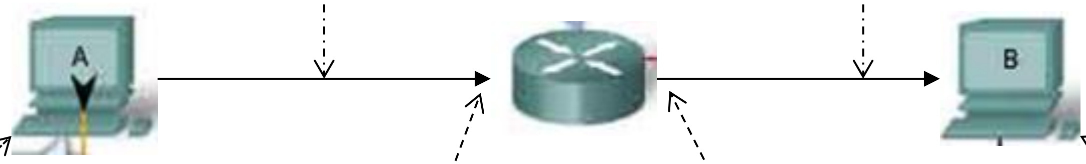
ARP Process through Router

Src MAC: 00-0d-88-c7-9a-24
Dst MAC: 00-0f-43-c5-8a-6
Src IP: 10.10.0.1
Dst IP: 10.20.0.1

Src MAC: 00-0e-36-c8-9c-06
Dst MAC: 00-08-a3-b8-ce-04
Src IP: 10.10.0.1
Dst IP: 10.20.0.1

Network 10.10.0.0

Network 10.20.0.0



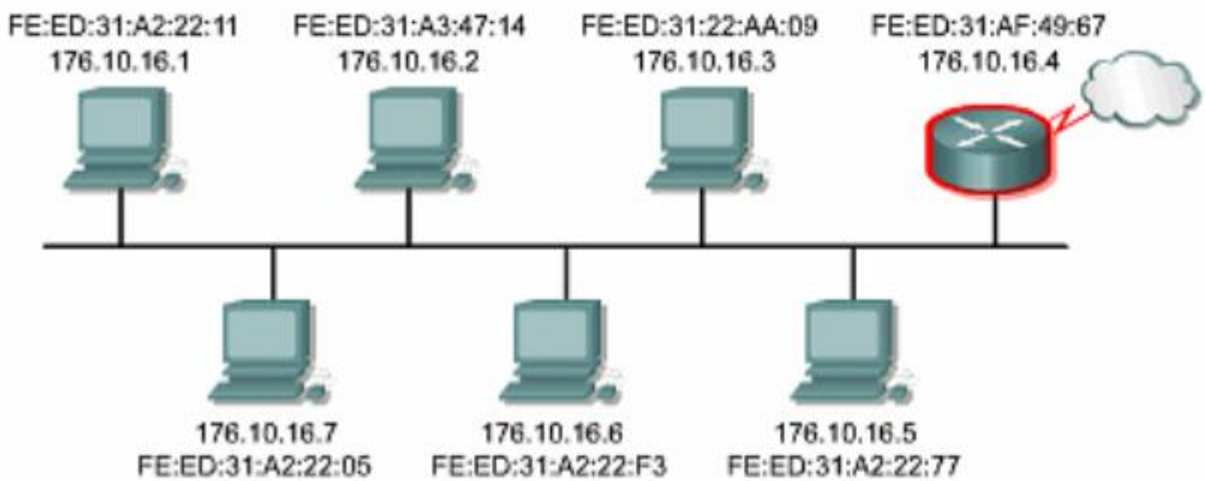
IP: 10.10.0.1/16
MAC: 00-0d-88-c7-9a-24

IP: 10.10.0.2/16
MAC: 00-0f-43-c5-8a-69

IP: 10.20.0.2/16
MAC: 00-0e-36-c8-9c-06

IP: 10.20.0.1/16
MAC: 00-08-a3-b8-ce-04

Proxy ARP and Default GW



ARP Table

IP Address	MAC Address
176.10.16.3	FE:ED:31:22:AA:09
176.10.16.6	FE:ED:31:A2:22:F3
176.10.16.5	FE:ED:31:A2:22:77
176.10.16.2	FE:ED:31:A3:47:14
199.11.20.5	FE:ED:31:AF:49:67

a) Proxy ARP

ARP Table

IP Address	MAC Address
176.10.16.3	FE:ED:31:22:AA:09
176.10.16.6	FE:ED:31:A2:22:F3
176.10.16.5	FE:ED:31:A2:22:77
176.10.16.2	FE:ED:31:A3:47:14
176.10.16.2	FE:ED:31:A3:47:14

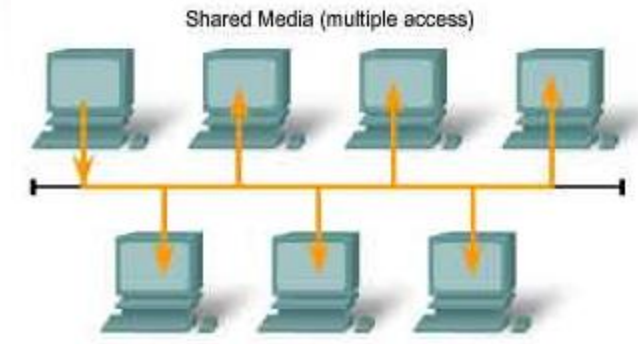
Default Gateway

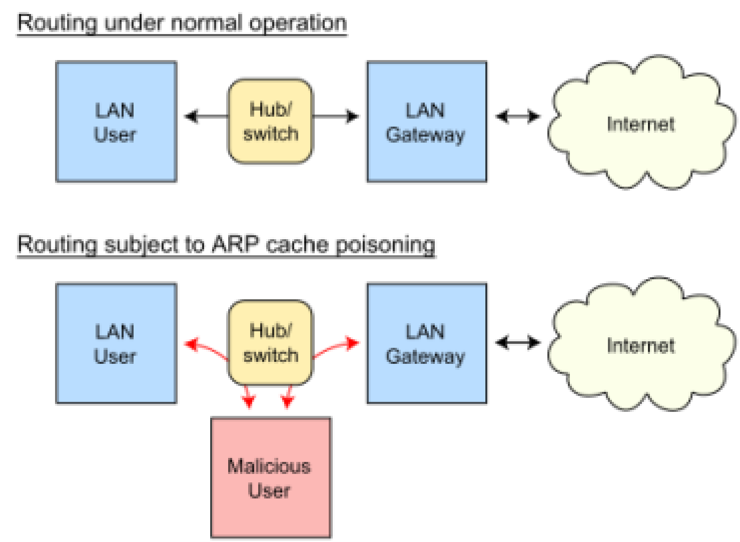
176.10.16.4	FE:ED:31:AF:49:67

b) Default Gateway

ARP Issues

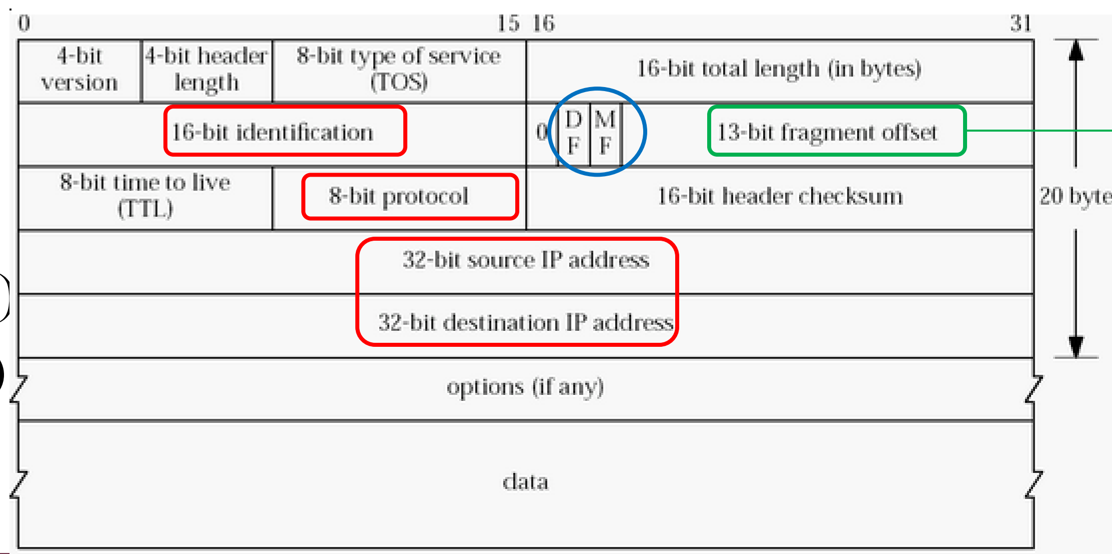
- Broadcasts
  - Overhead on the media
- Security
  - ARP Spoofing (ARP Poisoning)
    1. Attacker sends spoofed ARP onto a LAN
    2. Associate the attacker's MAC address with the IP address of a target (e.g., gateway)
    3. Attacker intercepts the packets and forward, modifies the data before forwarding, or launches a DOS attack





IP Fragmentation

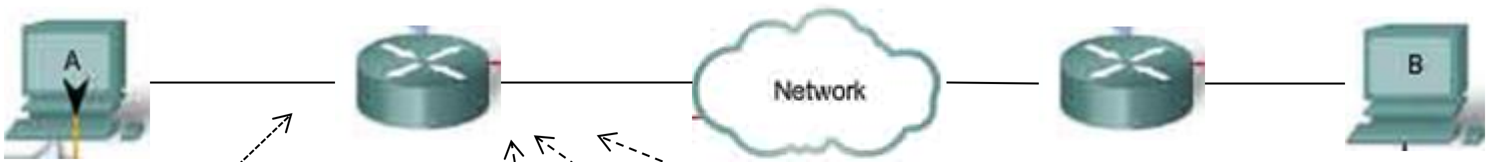
- Different networks different Maximum Transmission Unit (MTU).
  - Ethernet:1500 bytes, FDDI:4352 bytes, IEEE 802.11:7981 bytes
- The sending host or intermediate routers may fragment a datagram.
- The receiving host has responsibility for reassembly based on
  - same identification, protocol, source IP and destination IP.
- Flag
  - MF=1 (More Frag)
  - MF=0 (Last Frag)
  - DF=1 (Do not Frag)
  - DF=0 (Maybe frag)



Number of 8-byte chunks, not bytes

Fragmentation Process

MTU=1500 MTU=512 MTU=1500



Total Length=1500
ID: 26311
DF: 0
MF: 0
Offset: 0

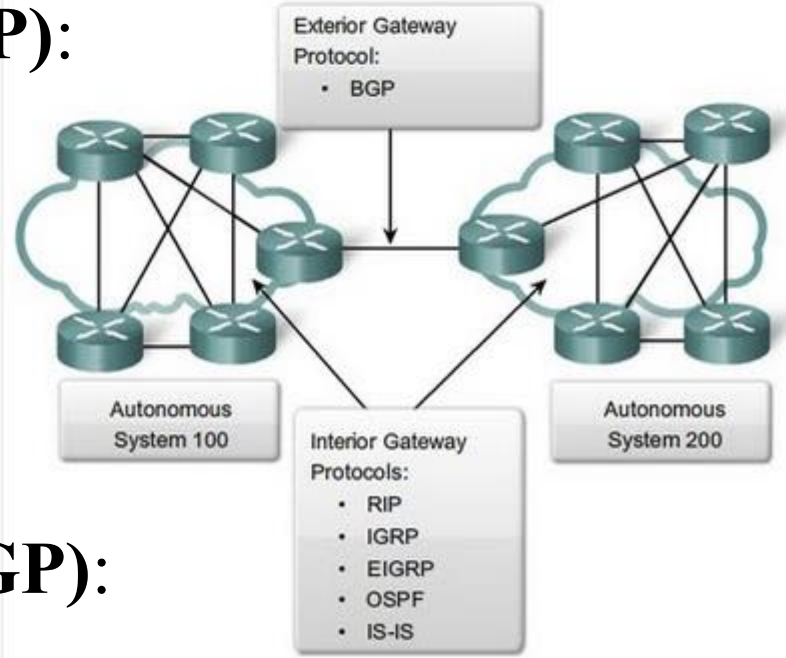
Fragment 1
Total Length=512
ID: 26311
DF: 0
MF: 1
Offset: 0

Fragment 2
Total Length=512
ID: 26311
DF: 0
MF: 1
Offset: 64

Fragment 3
Total Length=480
ID: 26311
DF: 0
MF: 0
Offset: 128

Dynamic Routing Protocol

- **Interior Gateway Protocols (IGP)**: for intra-domain routing
  - Distance Vector Protocols
    - RIP, IGRP, EIGRP
  - Link-State Protocols
    - OSPF, IS-IS
- **Exterior Gateway Protocols (EGP)**: for inter-domain routing
  - BGP

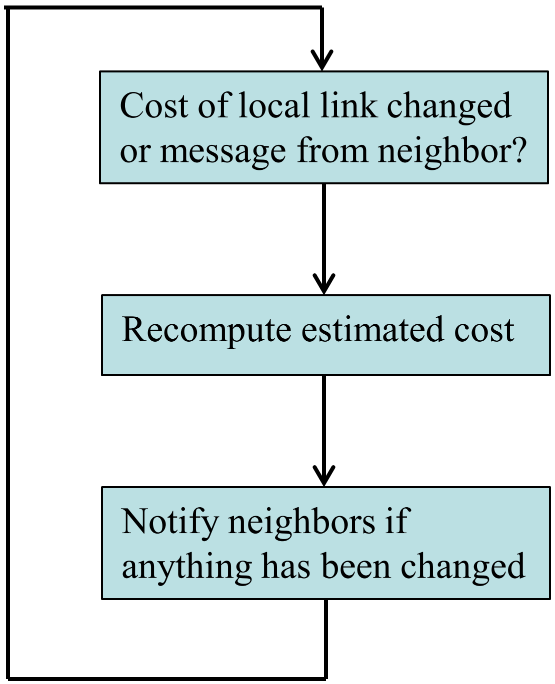


Autonomous System (AS)

- A **set of routers** under a single administrative domain,
  - operated within a uniform set of routing policies or common metrics
  - using an IGP to route packets within the AS
  - using an EGP to route packets to other ASes
- Example Autonomous Systems
  - A corporation might be an AS
  - A corporation with several different physical sites might be multiple ASes
  - An ISP might be an AS
  - An ISP might be multiple ASes

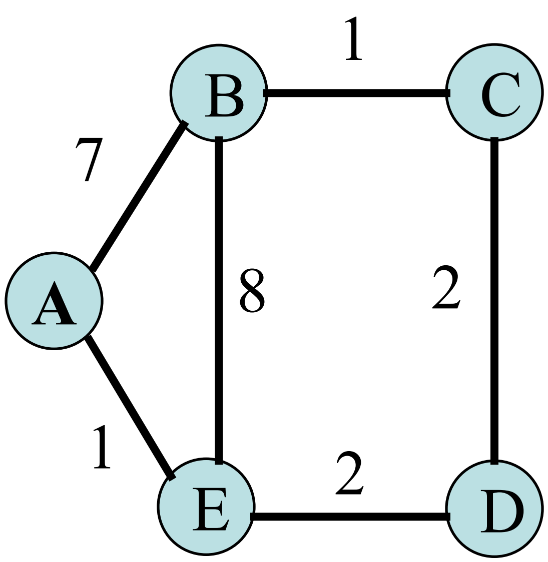
Distance Vector

- Each node constructs a vector containing the distances (costs) to all other nodes and distributes that vector to it's immediate neighbors.
  - Iterative, asynchronous: each local iteration caused by
    - Local link cost change
    - Distance vector update message from neighbor
  - Distributed
    - Each node notifies neighbors only when its DV changes
    - Neighbors then notify their neighbors if necessary.



– Tell your neighbors **when** you know about everyone.

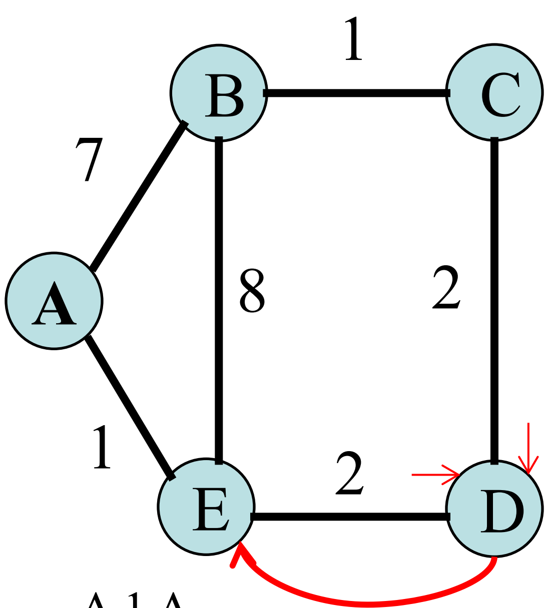
Distance Vector – Initial State



Distance to Reach Node

Info at node	**A**	**B**	**C**	**D**	**E**
A	0	7	∞	∞	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	∞	2	0

Distance Vector – D → E



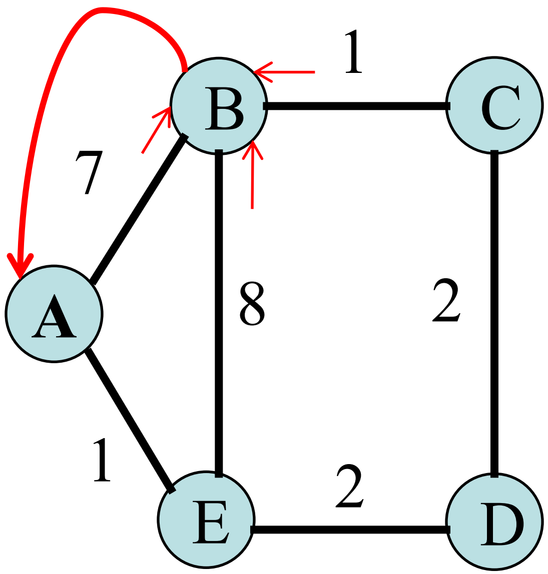
A 1 A
B 8 B
C **4 D**
D 2 D

	Distance to Reach Node				
Info at node	**A**	**B**	**C**	**D**	**E**
A	0	7	∞	∞	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	**4**	2	0

2+2 < ∞

Distance Vector – B → A

B 7 B
C **8 B**
D ∞
E 1 E



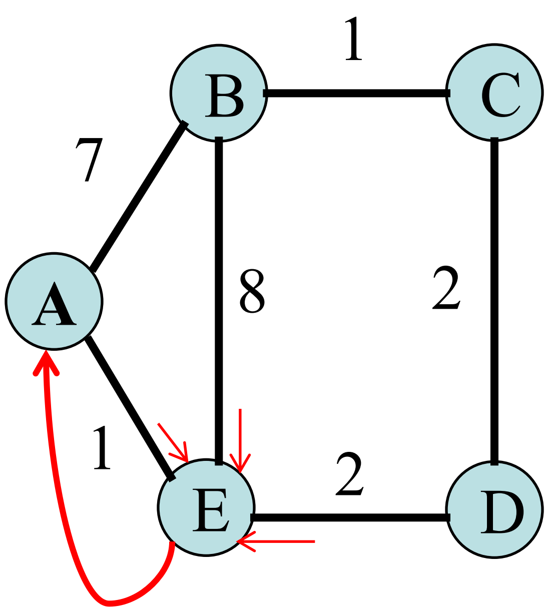
Distance to Reach Node

Info at node	A	B	C	D	E
A	0	7	**8**	∞	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	4	2	0

1+7 < ∞

Distance Vector – E → A

B 7 B
C **5 E**
D **3 E**
E 1 E

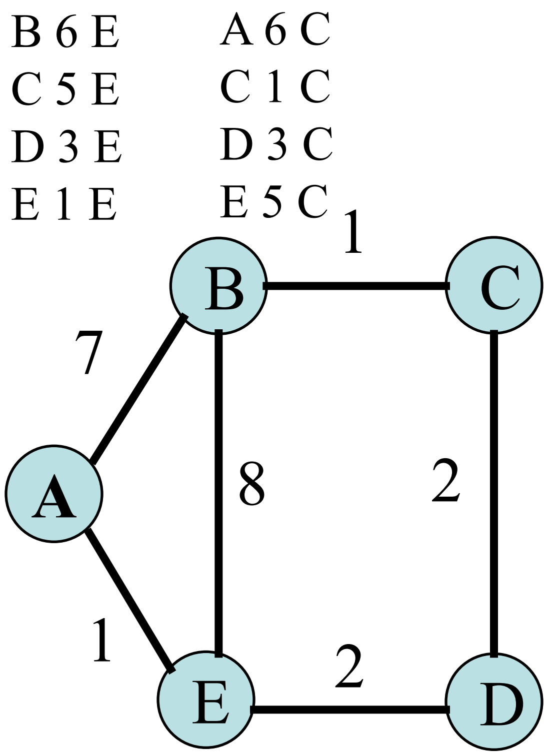


Distance to Reach Node

Info at node	A	B	C	D	E
A	0	7	**5**	**3**	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	4	2	0

$1+4 < 8$ **$1+2 < \infty$**

Distance Vector – after convergence



Distance to Reach Node

Info at node	**A**	**B**	**C**	**D**	**E**
A	0	6	5	3	1
B	6	0	1	3	5
C	5	1	0	2	4
D	3	3	2	0	2
E	1	5	4	2	0

RIP Characteristics

- Supports routing within an AS (IGP)
- Uses a **hop count metric**
- Uses a **Distance Vector Algorithm**
- The maximum number of hops allowed for RIP is 15 to prevent routing loops
- Each RIP router transmitted updates every 30s
- RIP uses the UDP and port number 520

RIP Advertisements

- Each router **broadcasts** copies of its routing table.
- Each entry identifies a **destination** as well as a **next hop** to that destination and a **cost** for the path
- Each router then **updates** its routing table based on improved routes advertised by other routers.

RIP Packet Format

<table>
<tr><td>command</td><td>version</td><td colspan="2">ZERO</td></tr>
<tr><td colspan="2">Family (net 1)</td><td colspan="2">Route Tag (net 1)</td></tr>
<tr><td colspan="4">IP address (net 1)</td></tr>
<tr><td colspan="4">Subnet mask (net 1)</td></tr>
<tr><td colspan="4">Next Hop (net 1)</td></tr>
<tr><td colspan="4">Distance – hop counts (net 1)</td></tr>
<tr><td colspan="2">Family (net 2)</td><td colspan="2">Route Tag (net 2)</td></tr>
<tr><td colspan="4">IP address (net 2)</td></tr>
<tr><td colspan="4">Subnet mask (net 2)</td></tr>
<tr><td colspan="4">Next Hop (net 2)</td></tr>
<tr><td colspan="4">Distance – hop counts (net 2)</td></tr>
</table>

Link State Algorithm

- Tell everyone what you know about your neighbors
- Two phases
  - Reliable flooding
    - Tell your routers what you know about your local topology
  - Path calculation (Dijkstra's Algorithm)
    - Finds the shortest path to each node
- Algorithm
  - Identifies the shortest path to nodes using 1 hop, then two hops, then 3 hops, etc. until values no longer change.

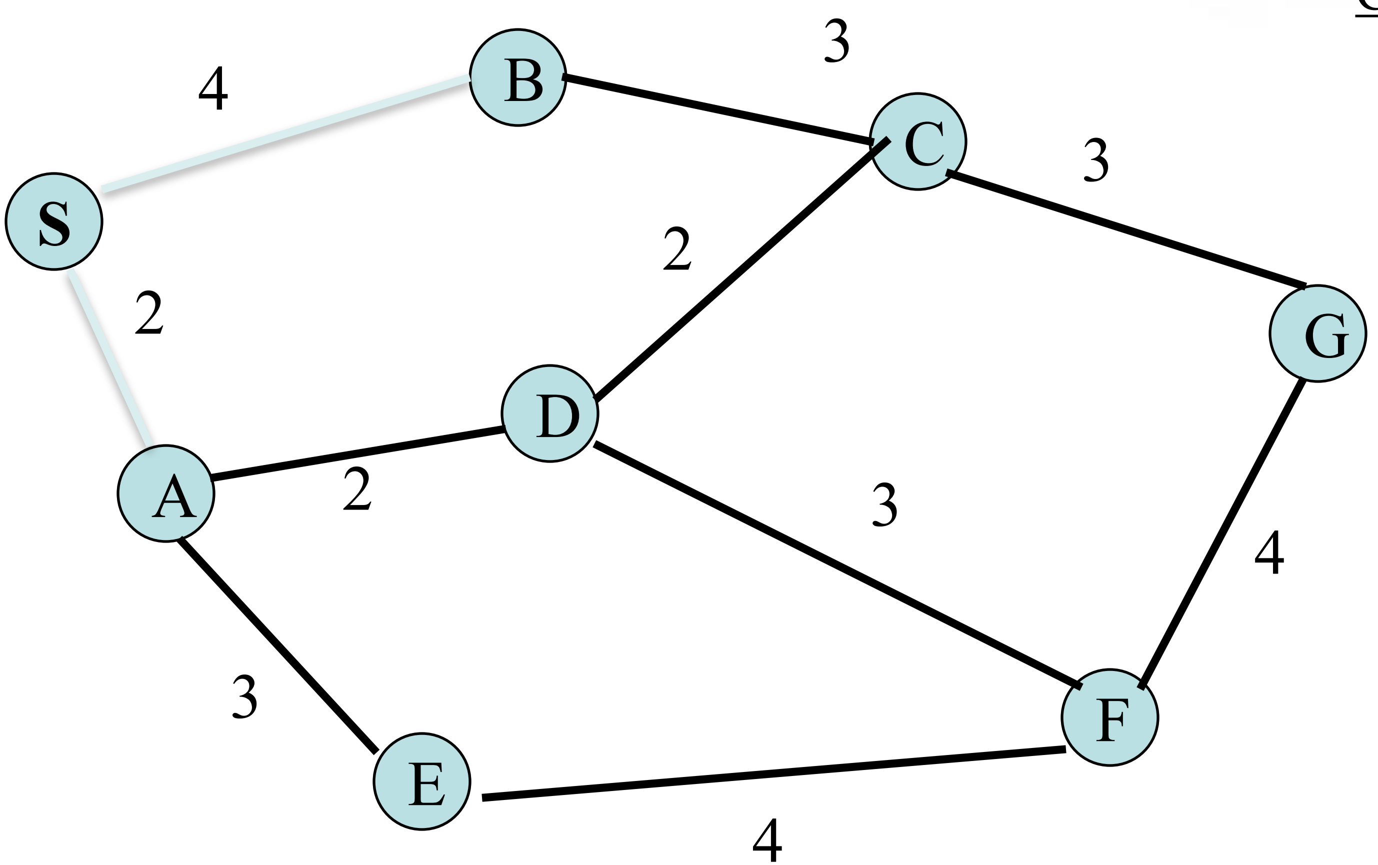
Open Shortest Path First (OSPF)

- Designed for Routing within an AS (IGP)
- Full CIDR and subnet Support
- Imported Routes supported
- Link-State Algorithm

Link State Advertisements

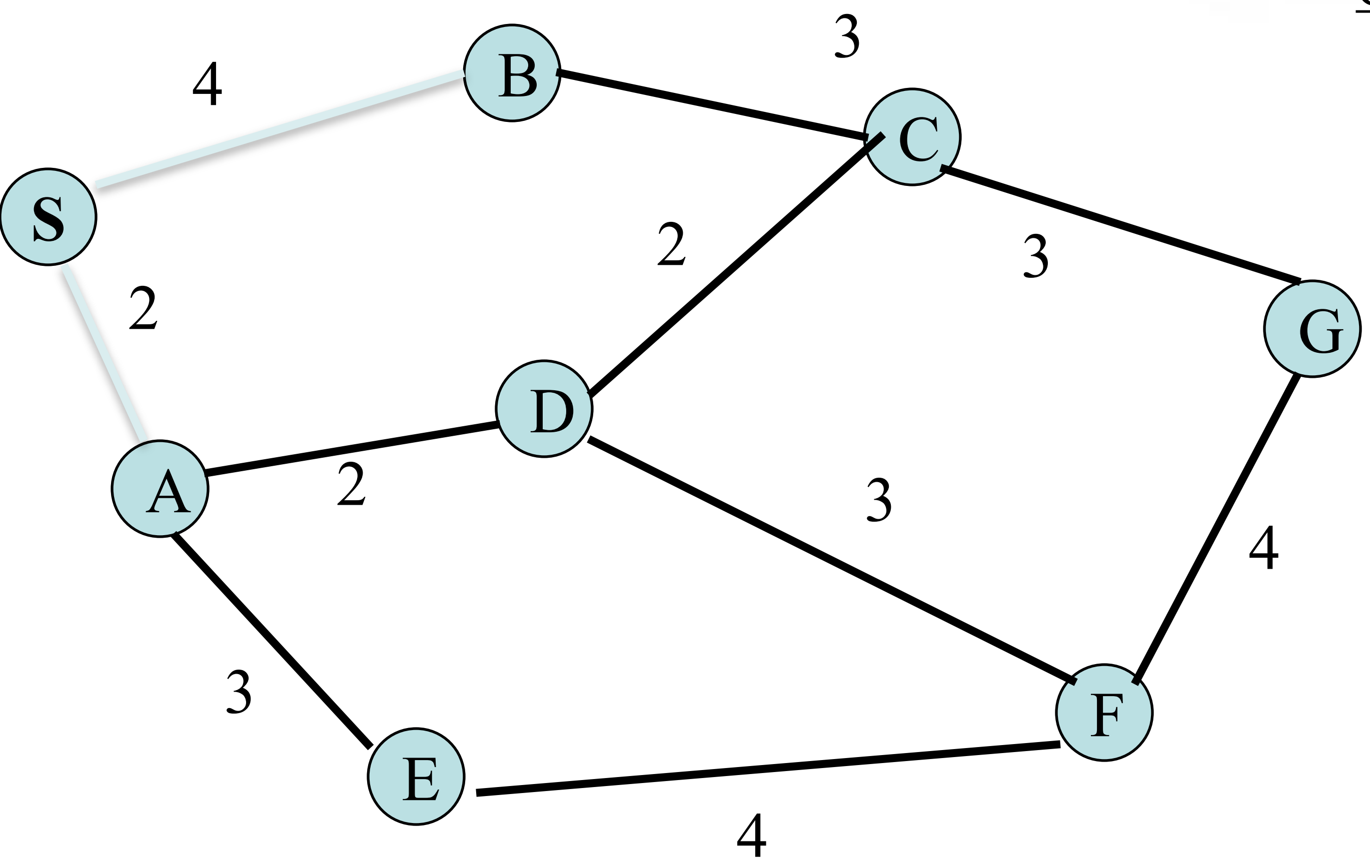
- Supports distributed routing table calculations
  - Each router learns about all network links and builds its own routing table
- LSA components
  - Identifier for source switch
  - List of links connected to that switch
  - Cost of each link

Link State Algorithm



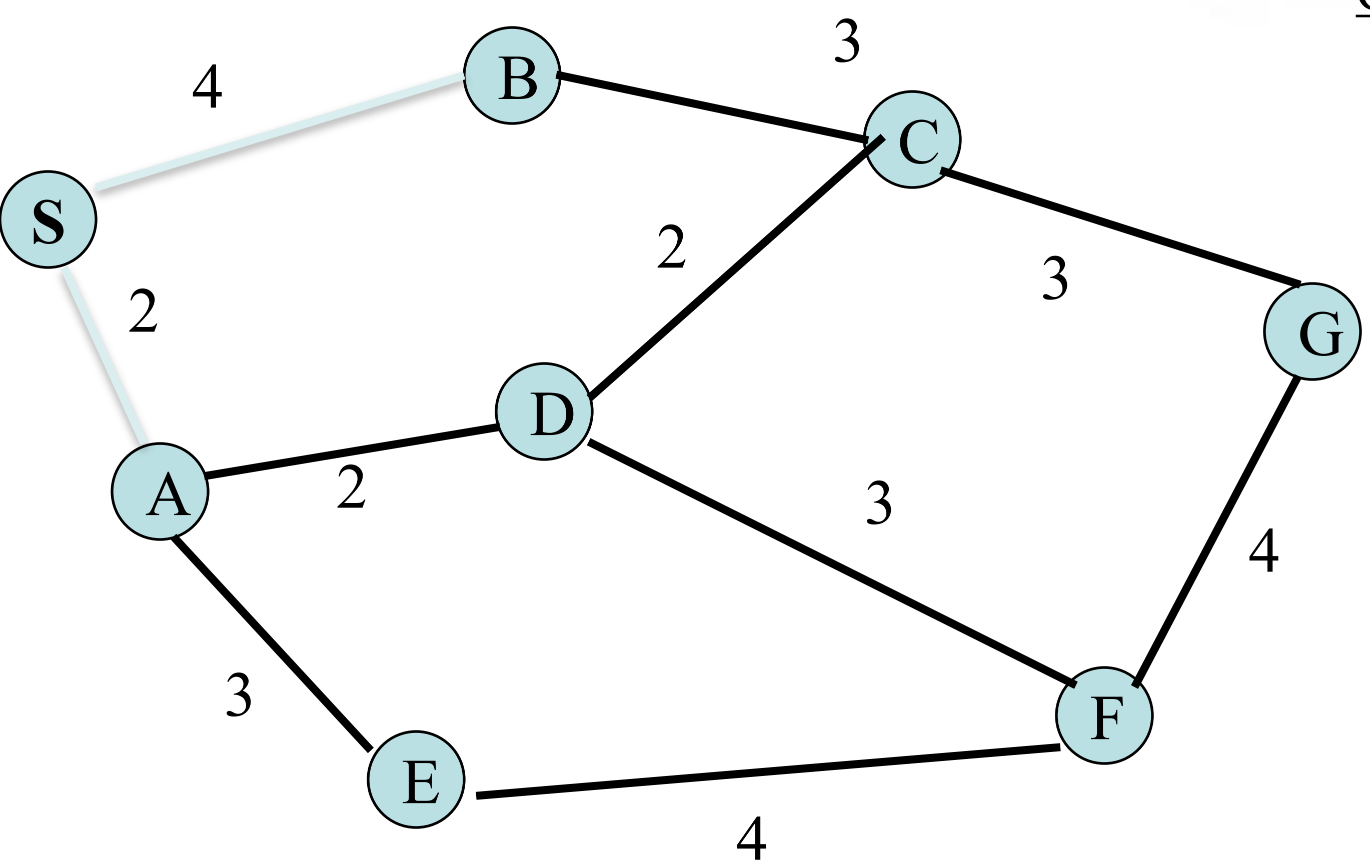
Confirmed	Tentative
S,0,-	A,2,A
	B,4,B

Link State Algorithm



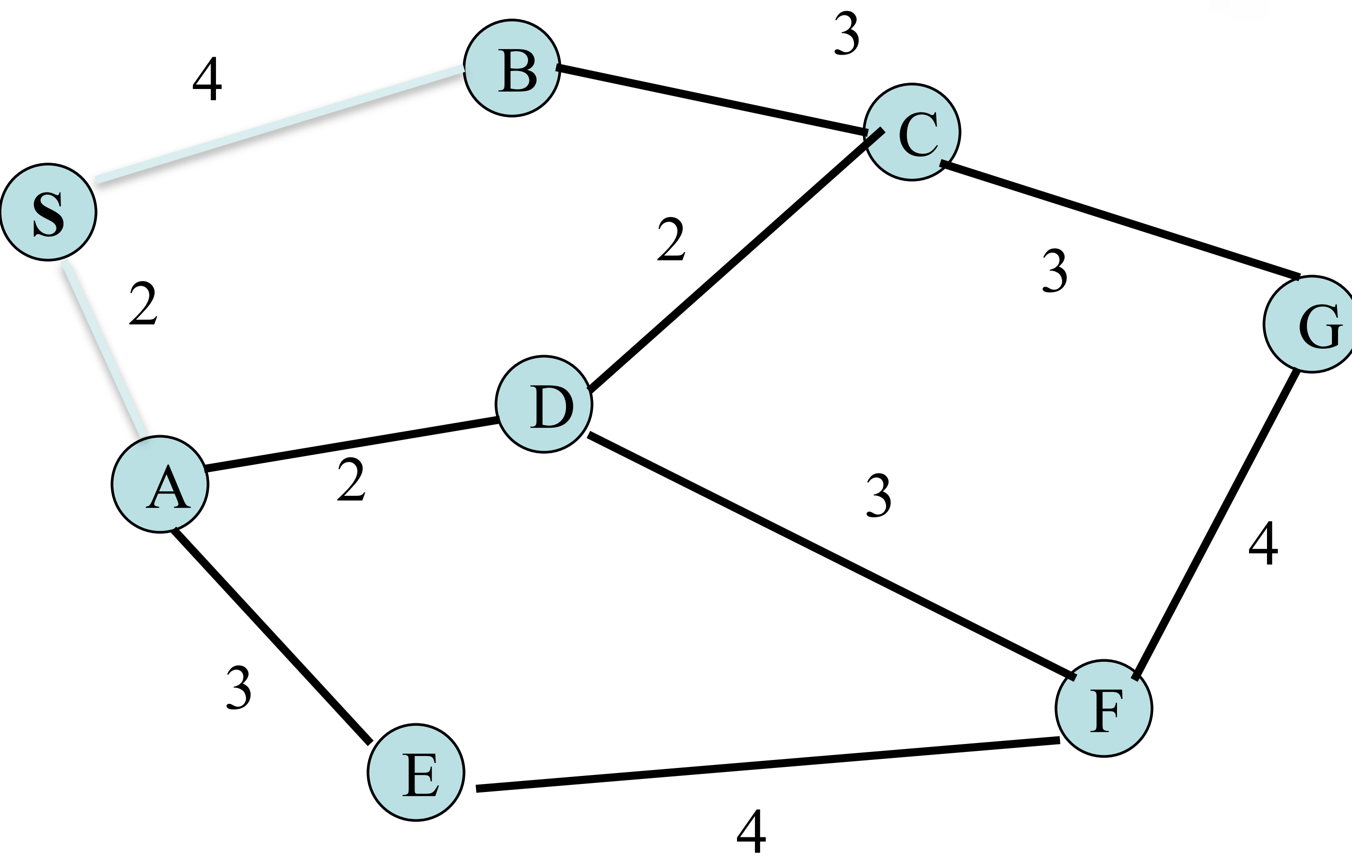
Confirmed	Tentative
S,0,-	B,4,B
A,2,A	D,4,A
	E,5,A

Link State Algorithm



Confirmed	Tentative
S,0,-	D,4,A
A,2,A	E,5,A
B,4,B	

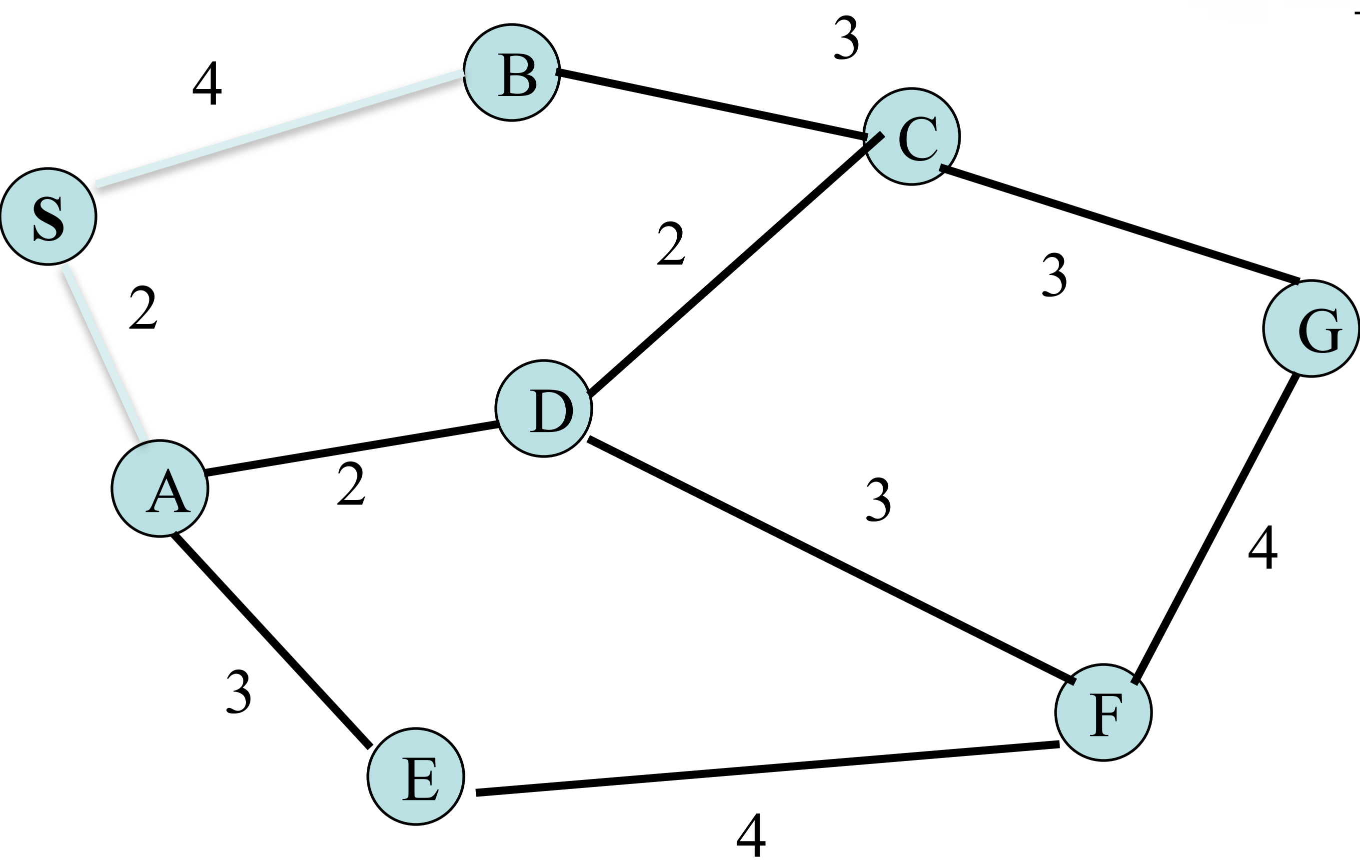
Link State Algorithm



Confirmed	Tentative
S,0,-	D,4,A
A,2,A	E,5,A
B,4,B	C,7,B

Link State Algorithm

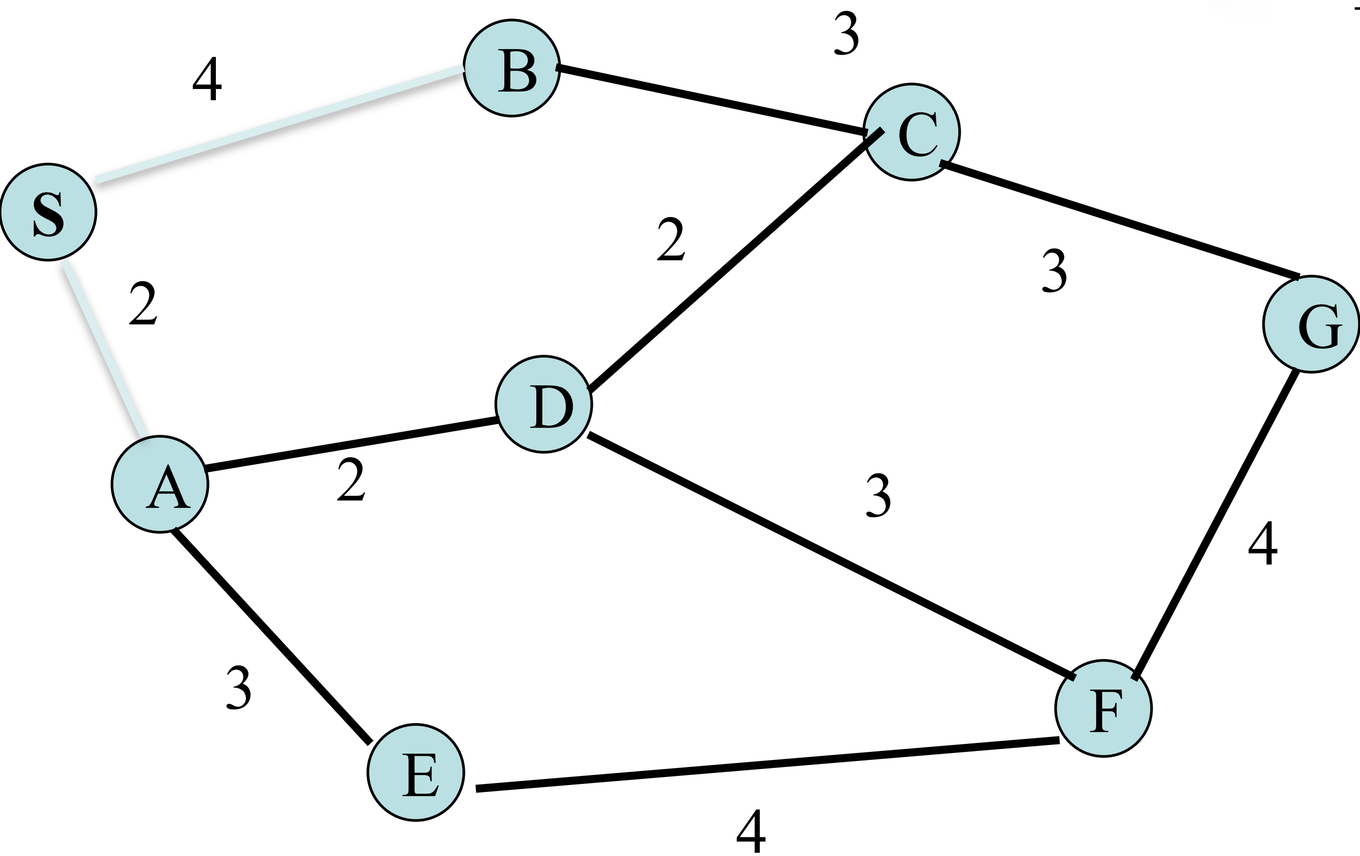
All 1 hop routes from Source



Confirmed	Tentative
S,0,-	E,5,A
A,2,A	C,7,B
B,4,B	
D,4,A	

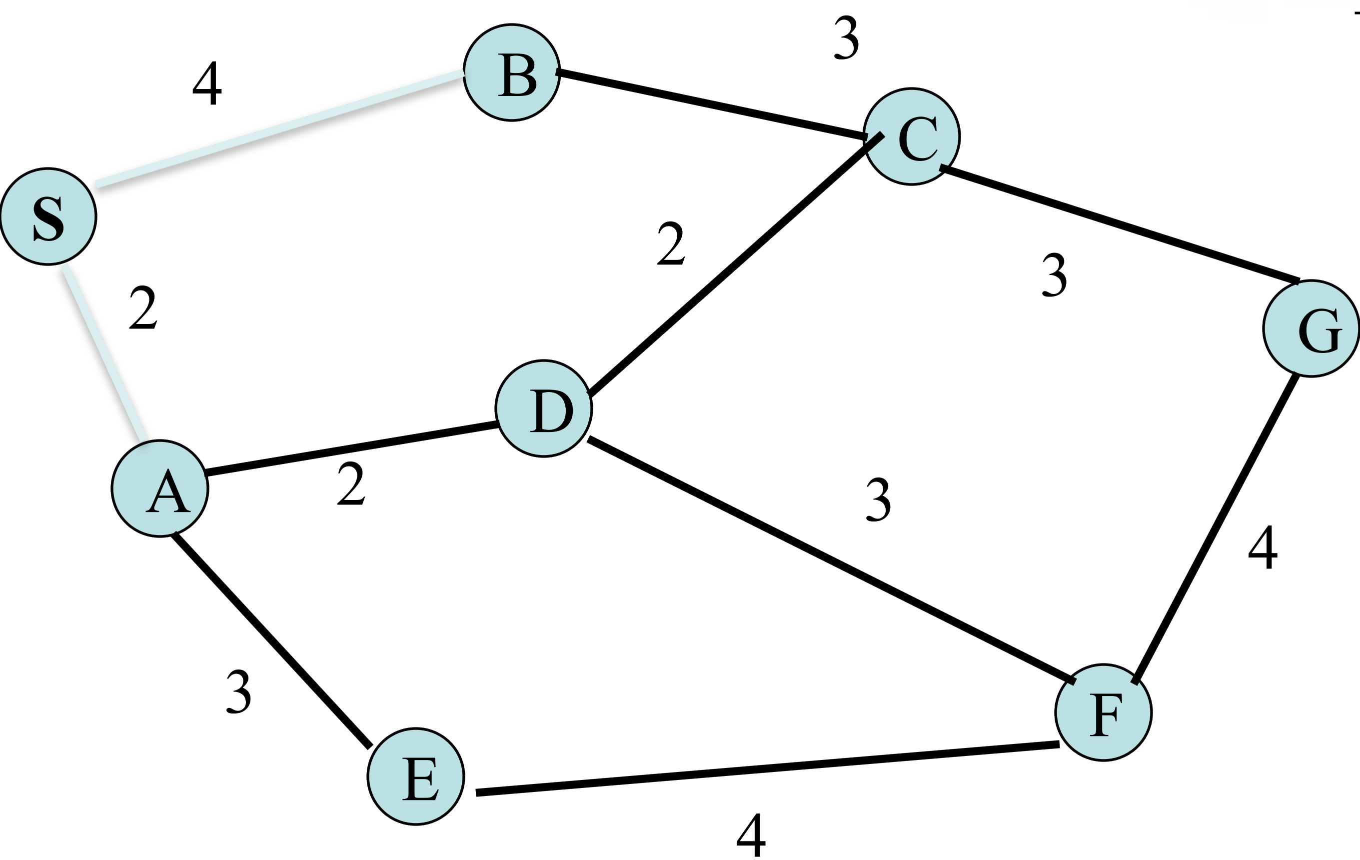
Link State Algorithm

All 1 hop routes from Source



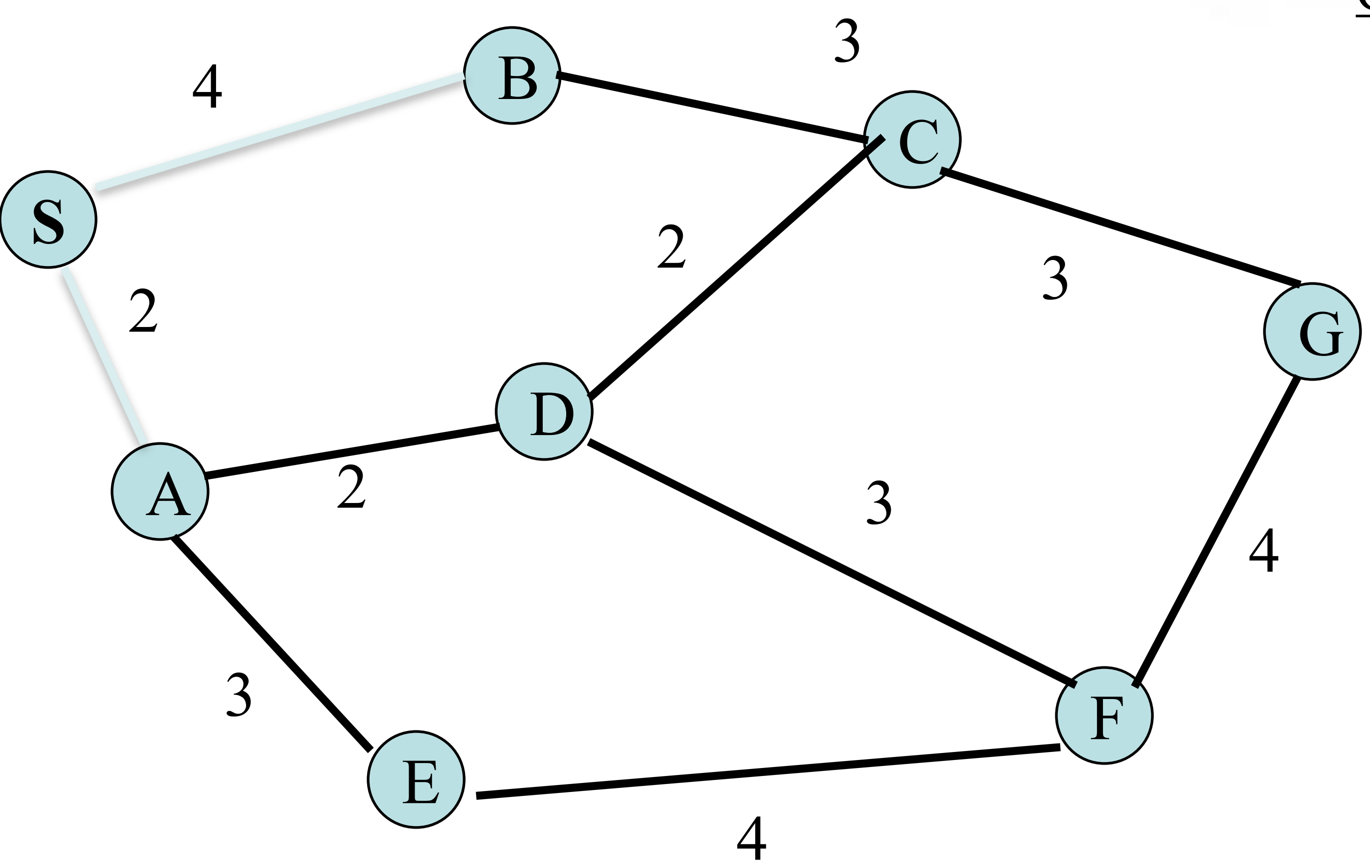
Confirmed	Tentative
S,0,-	E,5,A
A,2,A	C,6,D
B,4,B	F,7,D
D,4,A	

Link State Algorithm



Confirmed	Tentative
S,0,-	C,6,D
A,2,A	F,7,D
B,4,B	
D,4,A	
E,5,A	

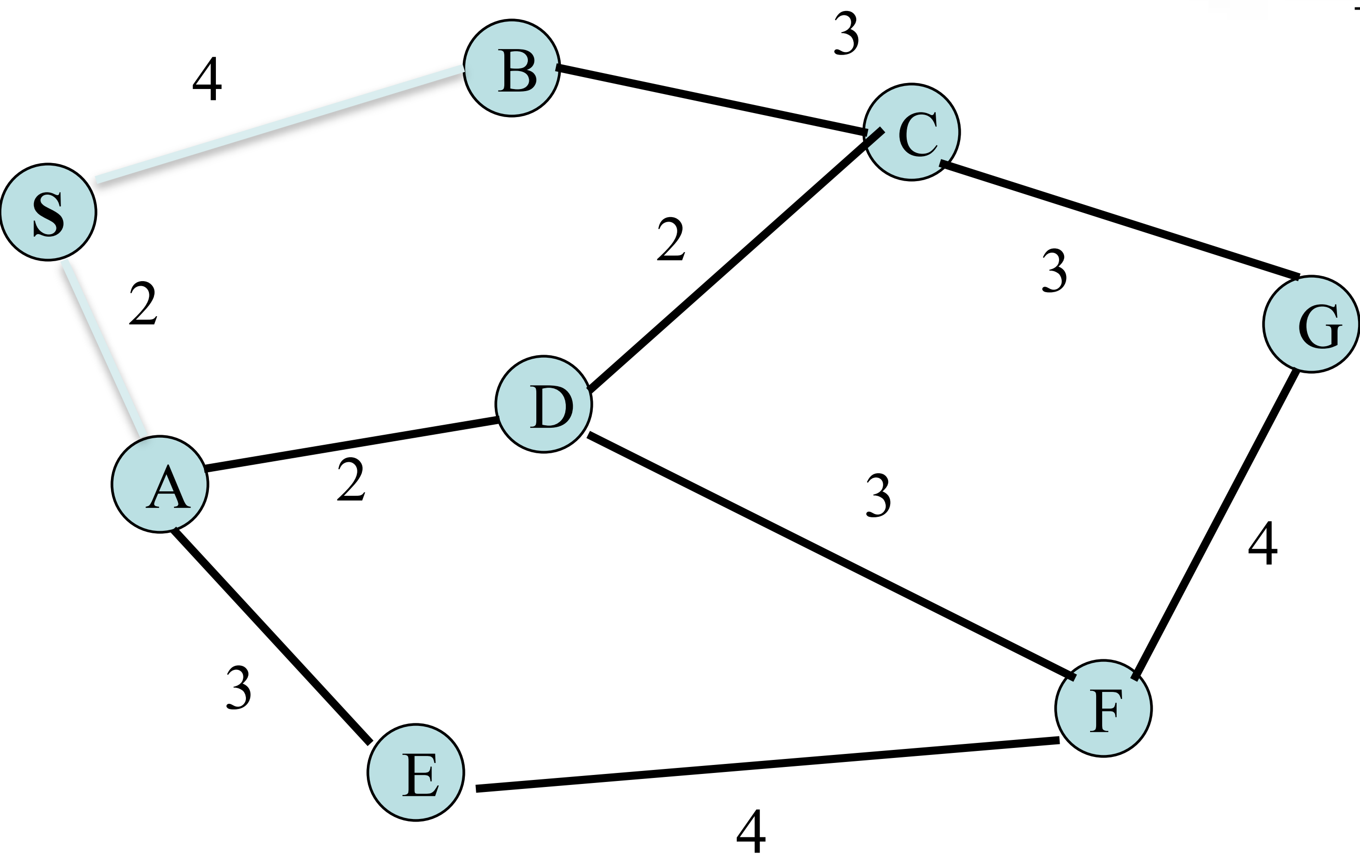
Link State Algorithm



Confirmed	Tentative
S,0,-	C,6,D
A,2,A	F,7,D
B,4,B	~~F,9,E~~
D,4,A	
E,5,A	

Link State Algorithm

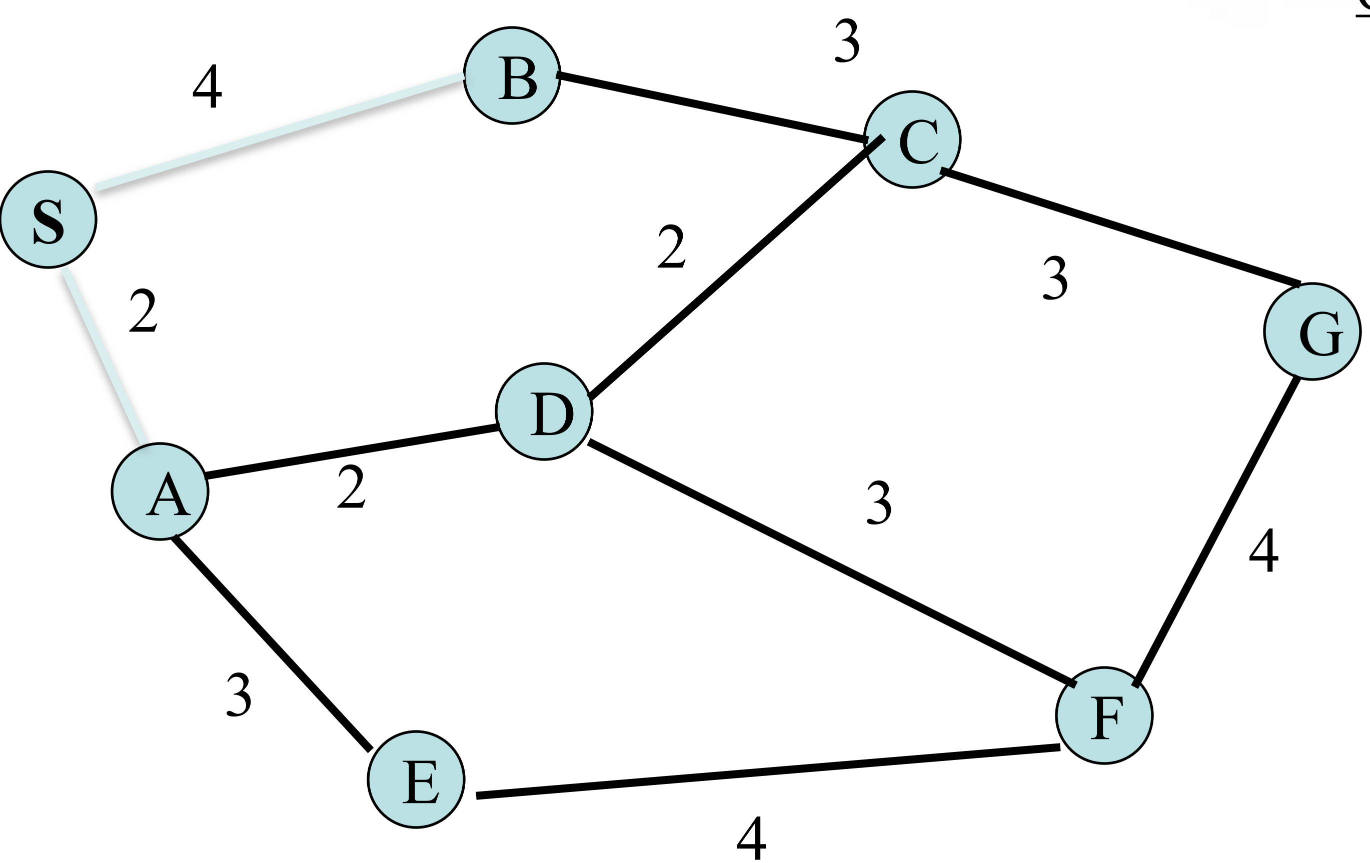
Best 1 and 2 hop routes from source



Confirmed	Tentative
S,0,-	F,7,D
A,2,A	
B,4,B	
D,4,A	
E,5,A	
C,6,D	

Link State Algorithm

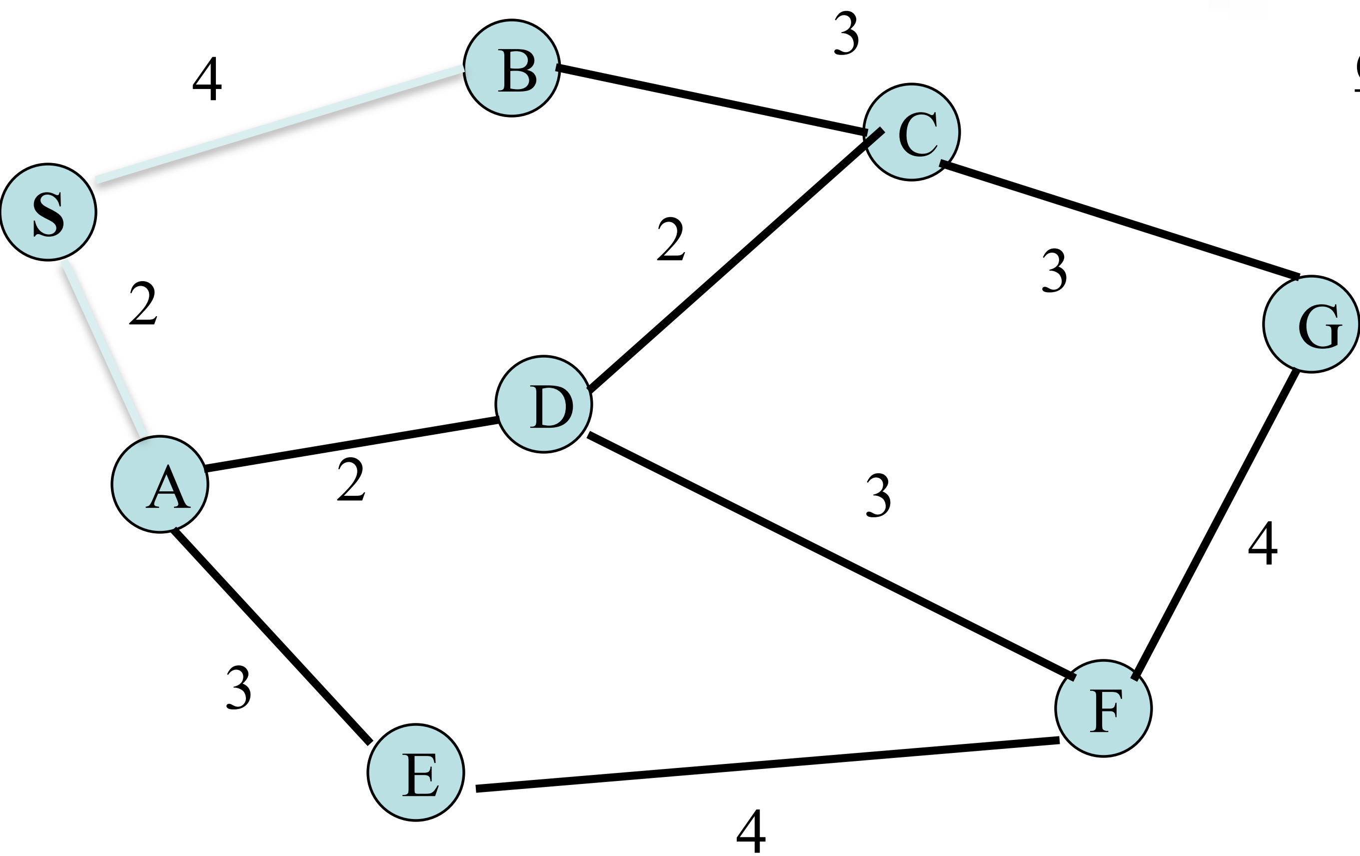
Best 1 and 2 hop routes from source



Confirmed	Tentative
S,0,-	F,7,D
A,2,A	G,9,C
B,4,B	
D,4,A	
E,5,A	
C,6,D	

Link State Algorithm

Best 1, 2, and 3 hop routes from source: All nodes connected.



Confirmed	Tentative
S,0,-	
A,2,A	
B,4,B	
D,4,A	
E,5,A	
C,6,D	
F,7,D	
G,9,C	

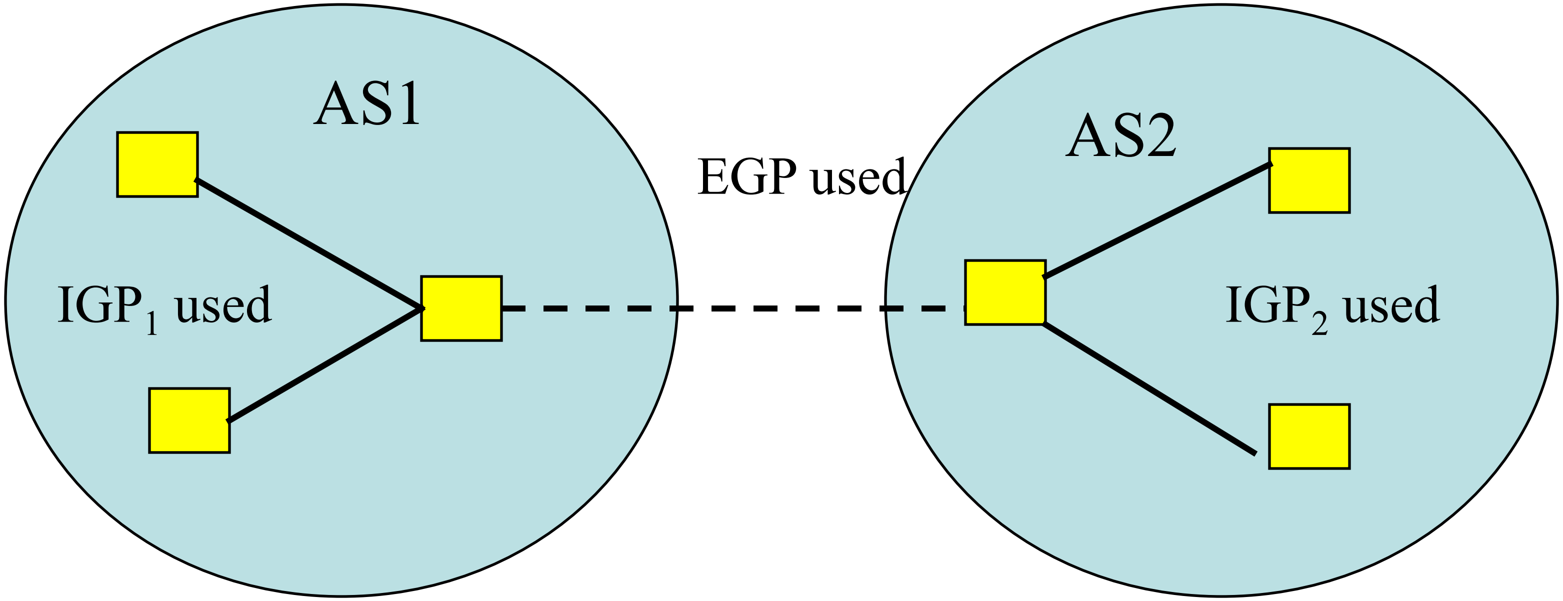
Distance Vector vs. Link State

- Both protocols are designed to find shortest paths between routers
- RIP passes more information, but scales poorly (network traffic is $O(n^2)$)
- OSPF requires more work of router, but scales well
- RIP optimizes total network (link) cost
- OSPF finds shortest paths to other nodes

IGP vs. EGP

- RIP and OSPF are Interior Gateway Protocols (route within an AS)
- The Internet can be modeled as a set of arbitrarily connected Autonomous Systems
- We need some other protocol (EGP) to route packets between ASs.

IGP vs. EGP



AS-AS Traffic

- Traffic types:
  - Local traffic: Traffic that originates or terminates in the local AS
  - Transit traffic: Traffic that passes through an AS
- AS types:
  - **Stub AS**: Only one connection to another AS
  - **Multihomed AS**: multiple AS connections
  - **Transit AS**: a Multihomed as what supports transit traffic.

Exterior Gateway Protocols

- Identifies routes in terms of a series of Autonomous Systems.
- The sequence of ASs must represent a series of exterior gateway neighbors (machines that communicate directly with each other).
- Routing **within** an AS is not identified.

Border Gateway Protocol

- Can be used to route among Autonomous Systems (EGP)
- Supports the use of policies in terms of which routes are advertised outside the AS
- Provides facilities for Transit Routing
- Uses Reliable Transport (TCP)
- Supports the use of performance related criteria to evaluate the quality of different routes

Summary

- Discussed routing tables
- Interior Gateway Protocols
  - RIP (Distance Vector)
  - OSPF (Link State)
- Exterior Gateway Protocols
  - BGP