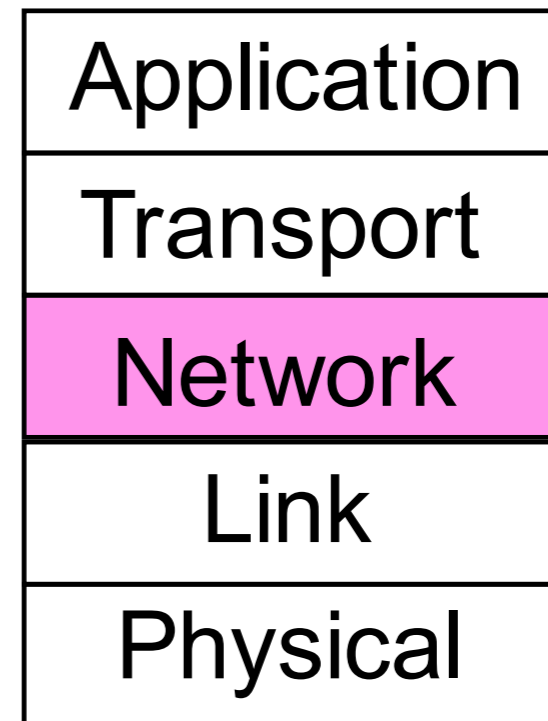


# Network Layer

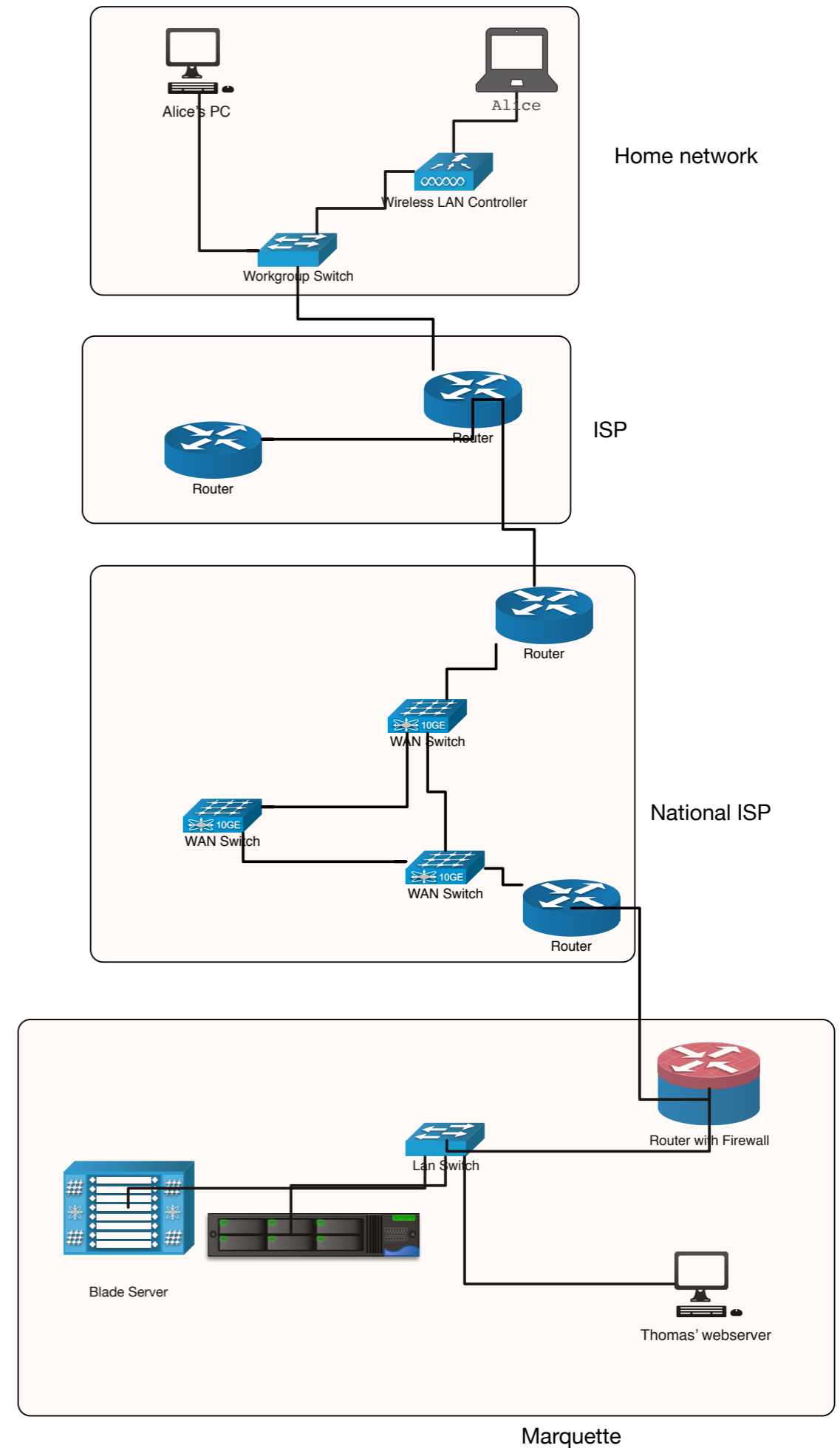
Networking  
Marquette University, 2022

# Network Layer

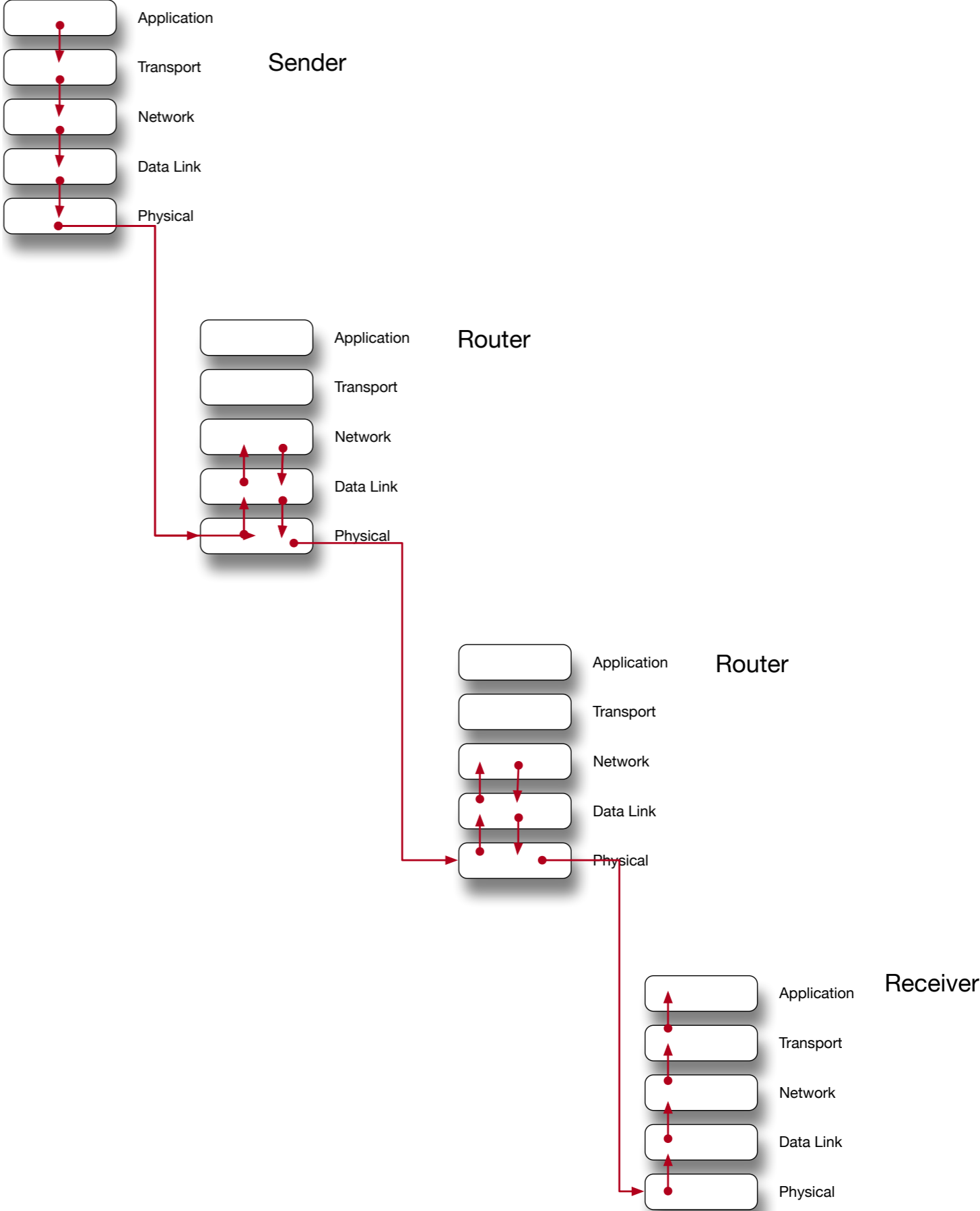
- Responsible for
  - Delivering packets
    - between endpoints
    - over multiple links



# Networking Layer



# Networking Layer

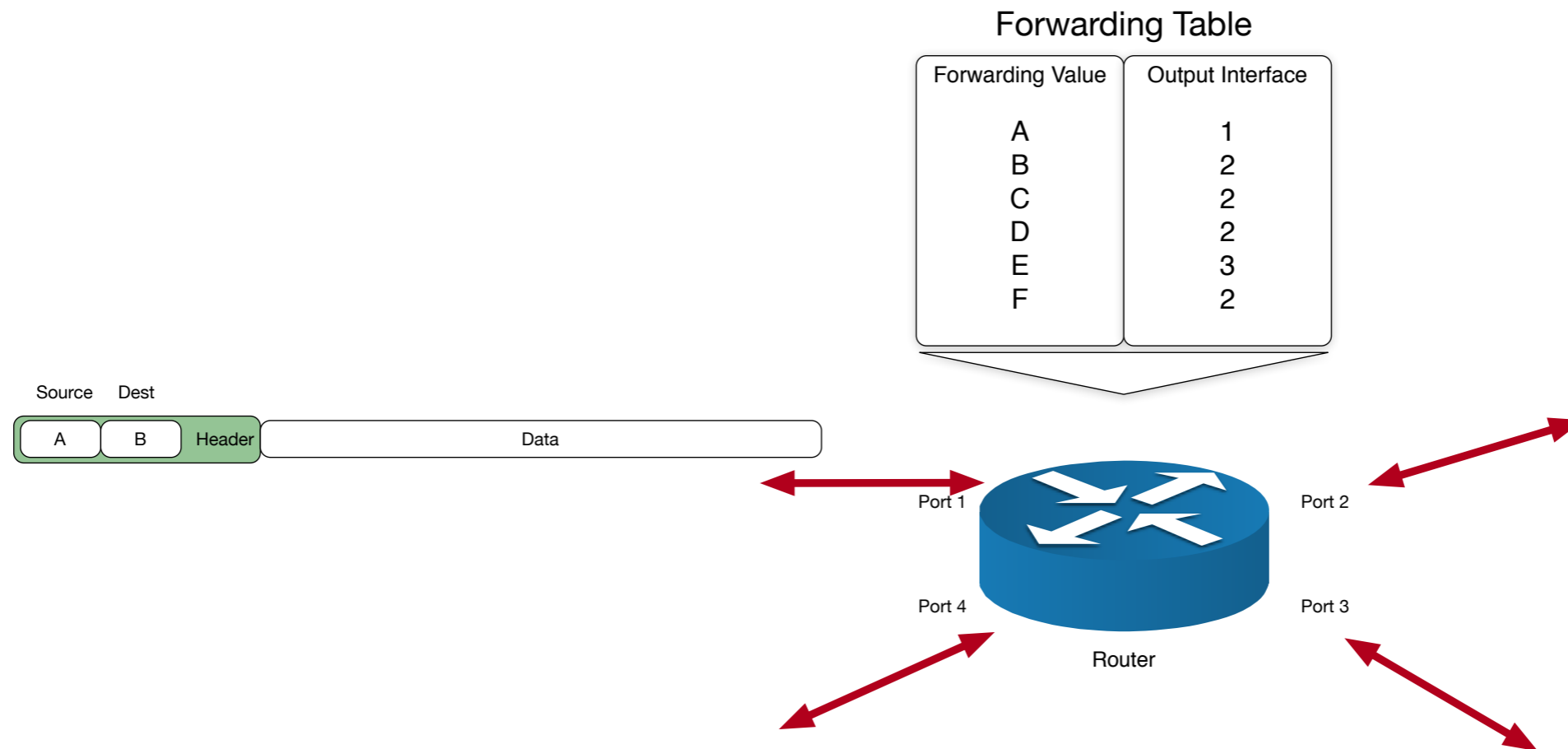




# Network Layer Services

- Packetizing
- Routing and Forwarding
  - Routing: Find the *best* route with routing protocol
  - Forwarding: What happens when a router receives a package
- Error Control
  - not provided in the TCP/IP stack, but ICMP provides similar services
- Flow Control
  - not provided in the TCP/IP stack
- Congestion Control
- Quality of Service
- Security

# Forwarding

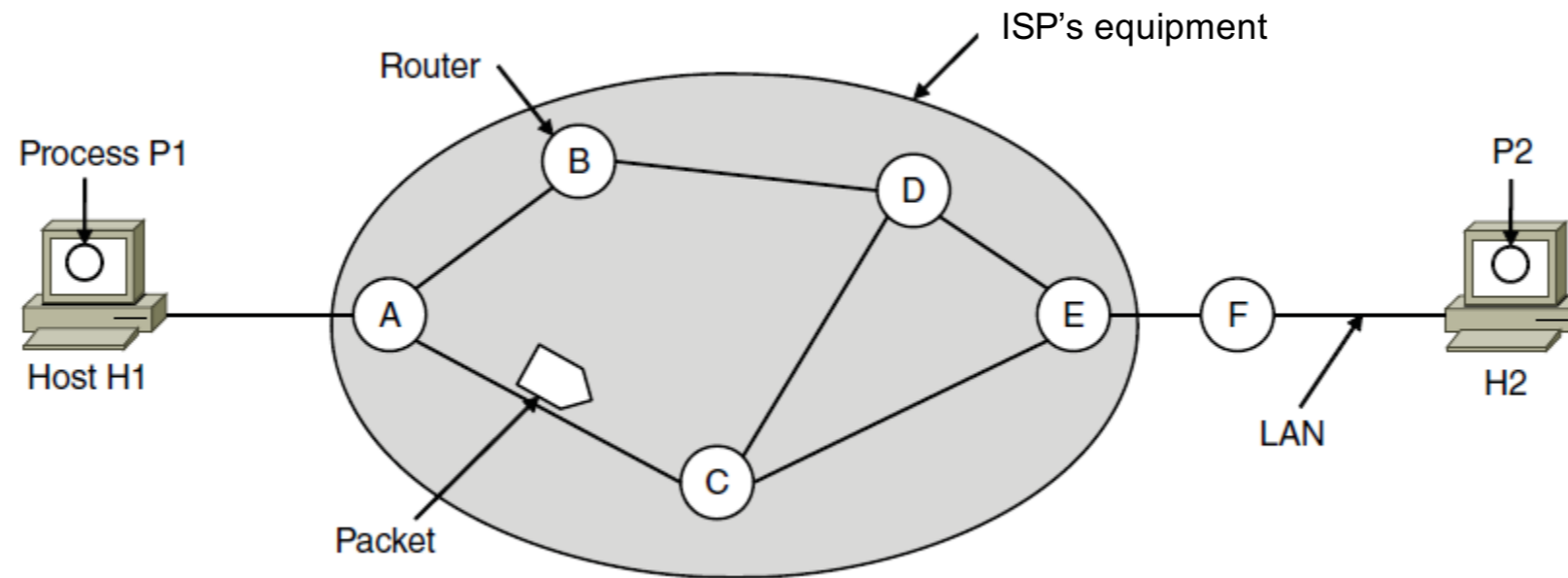


- Routers use a forwarding table
  - Routing protocols populate forwarding tables
  - Forwarding tables needs to be compacted

# Packet Switching

- Datagram Approach: Connectionless Service
  - Each packet is routed according to the destination value in the header
- Virtual Circuit Approach: Connection-oriented Service
  - All packets belonging to a message contain a virtual connection identifier
  - First a connection is set up by sending a request package and an ack package
  - Then all routers on the way use the connection identifier to route the package.

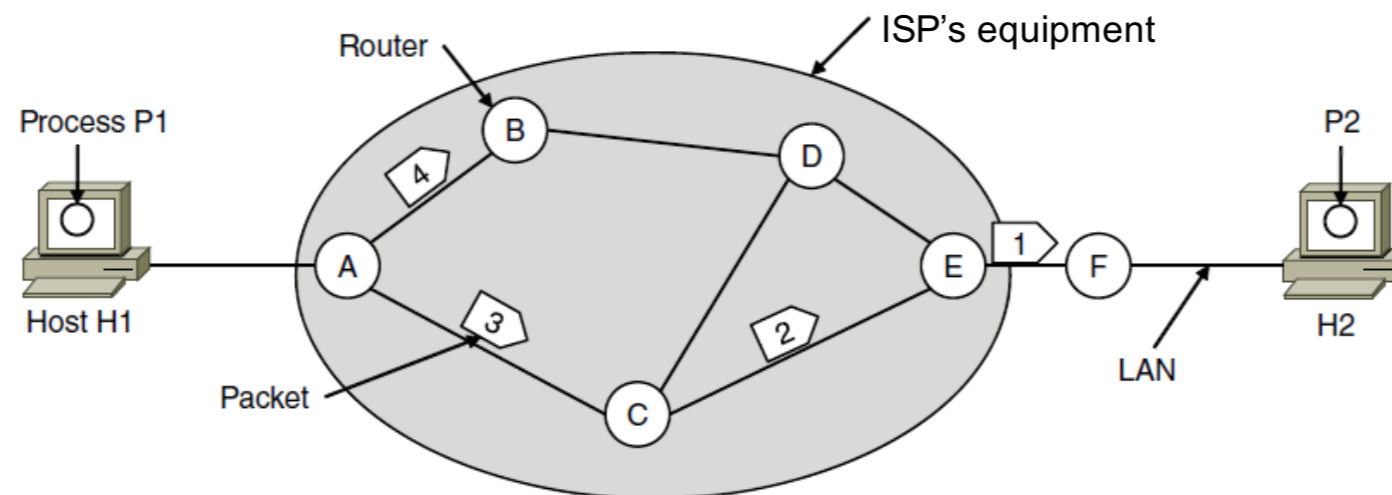
# Store and Forward Packet Switching



- Hosts send packets into the network
- Packets are forwarded by routers

# Connectionless Services — Datagrams

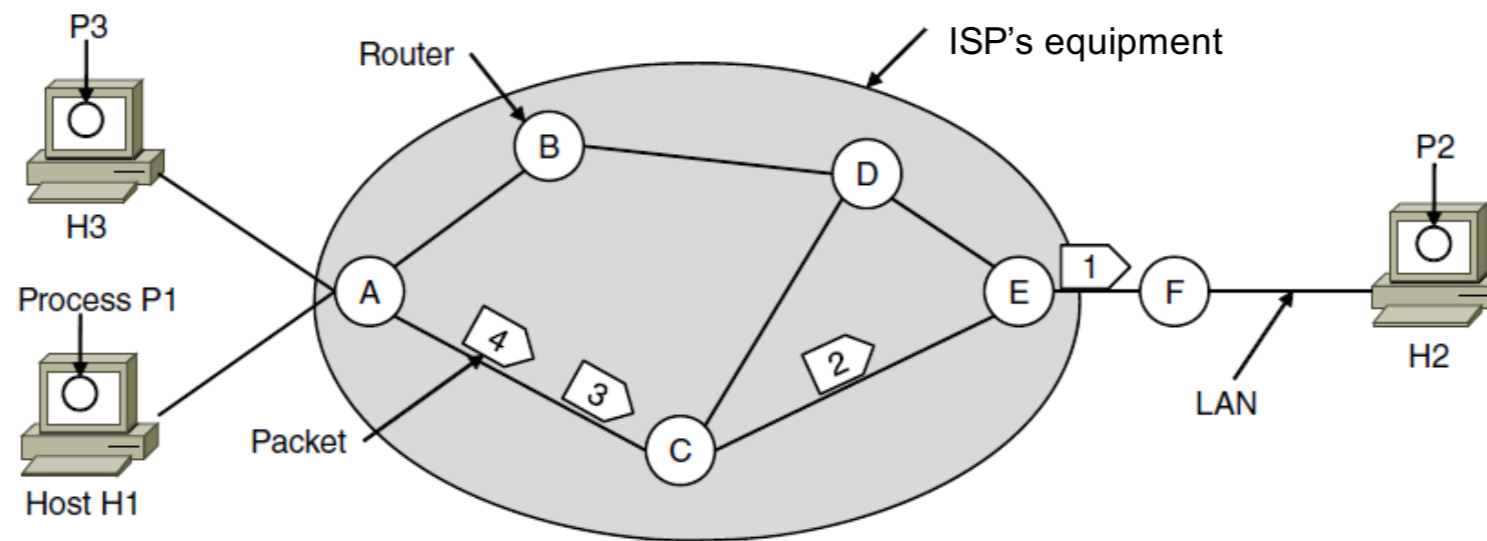
- Packet is forwarded using destination address in it
- Different packets can take different paths



A's table (initially)	A's table (later)	C's Table	E's Table
A	A	A	A
B	B	B	B
C	C	C	C
D	D	D	D
E	E	E	E
F	F	F	F
Dest.	Line		

# Connection-Oriented Virtual Circuits

- Packet is forwarded along a virtual circuit using a tag inside
- Virtual circuit is set up ahead of time



A's table

H1	1	C	1
H3	1	C	2
In		Out	

C's Table

A	1	E	1
A	2	E	2

E's Table

C	1	F	1
C	2	F	2

# Comparison

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

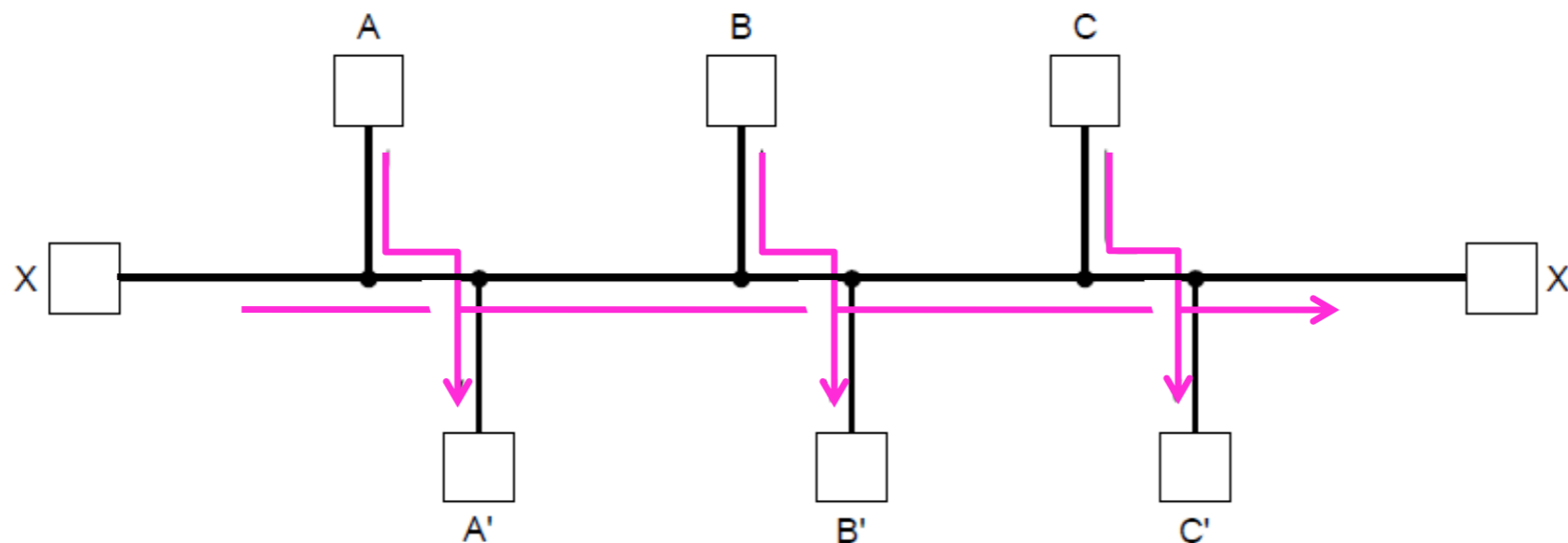
# Network Layer Performance

- Delay:
  - Transmission Delay
  - Propagation Delay
  - Processing Delay
  - Queuing Delay
- Throughput
- Packet loss
- Congestion control



# Routing Algorithms

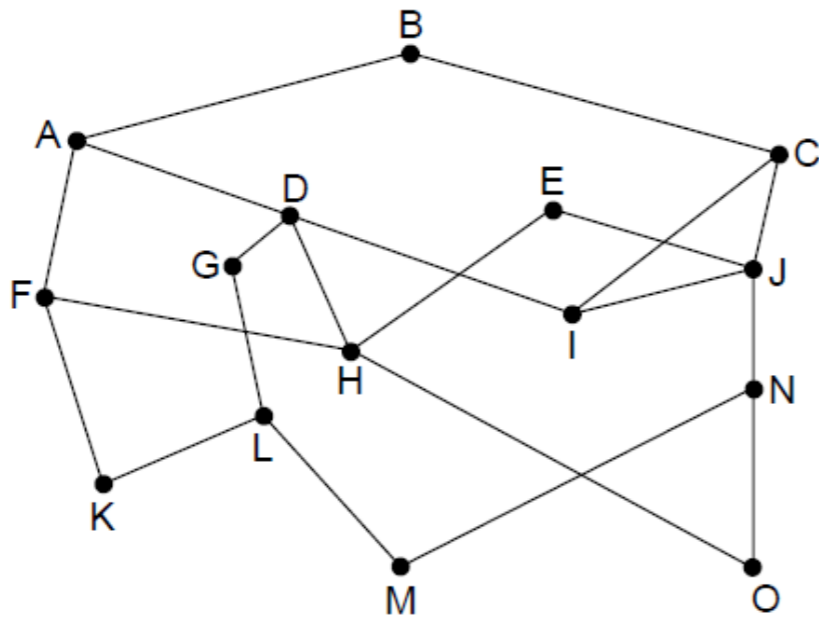
- Routing is the process of discovering network paths
  - Model the network as a graph of nodes and links
  - Decide what to optimize (e.g., fairness vs efficiency)
  - Update routes for changes in topology (e.g., failures)
- Forwarding is the sending of packets along a path



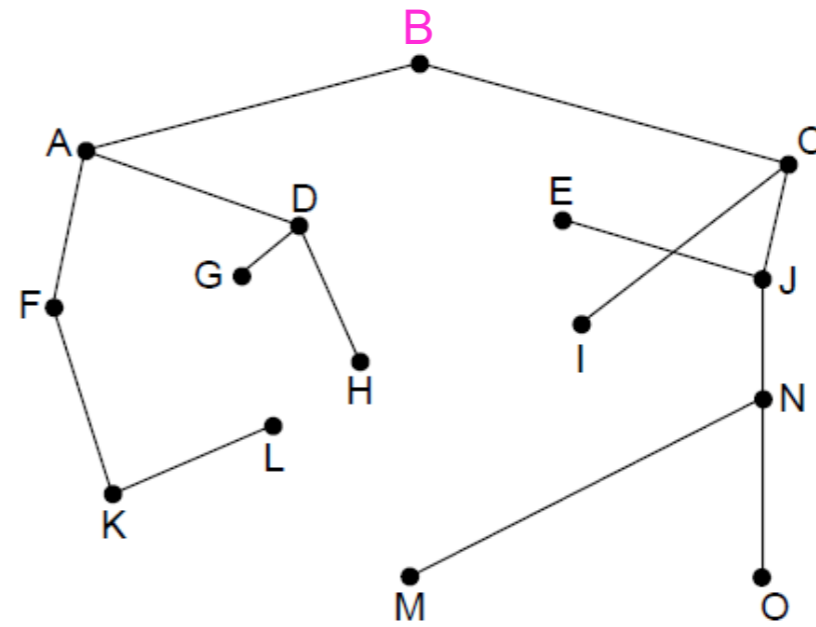
Network with inherent fairness vs. efficiency conflict

# The Optimality Principle

- Each portion of a best path is also a best path; the union of them to a router is a tree called the sink tree
- Best means fewest hops in the example



Network



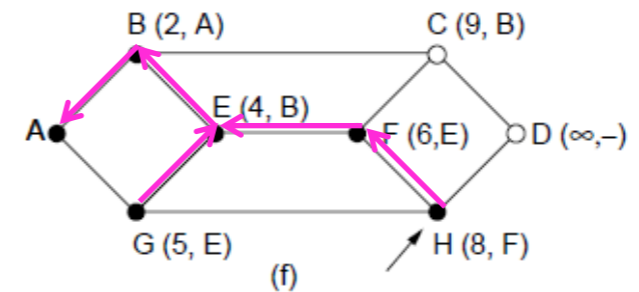
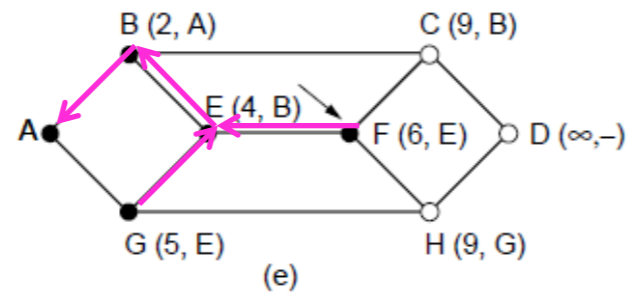
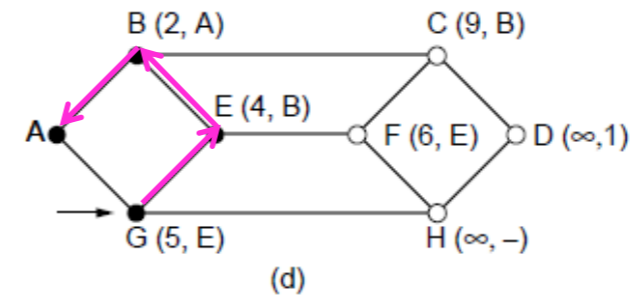
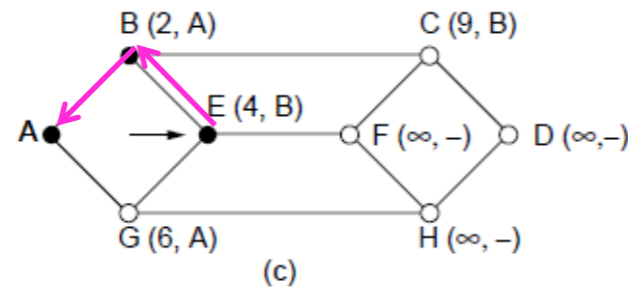
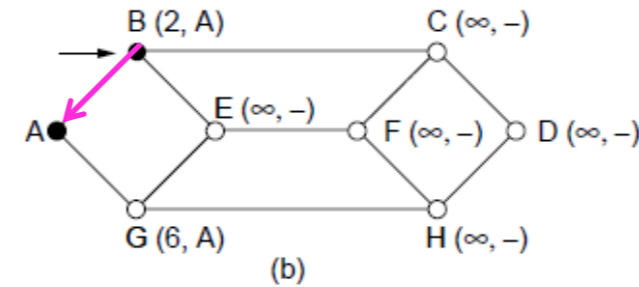
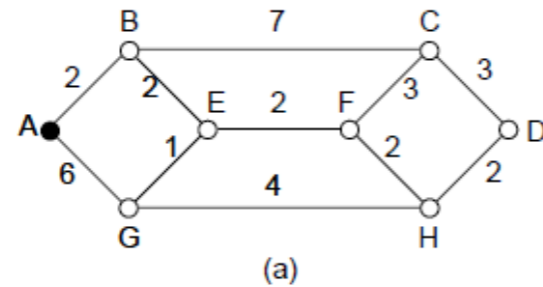
Sink tree of best paths to router B

# Shortest Path Algorithm

- Dijkstra's algorithm computes a sink tree on the graph:
  - Each link is assigned a non-negative weight/distance
  - Shortest path is the one with lowest total weight
  - Using weights of 1 gives paths with fewest hops
- Algorithm:
  - Start with sink, set distance at other nodes to infinity
  - Relax distance to other nodes
  - Pick the lowest distance node, add it to sink tree
  - Repeat until all nodes are in the sink tree

# Shortest Path Algorithm

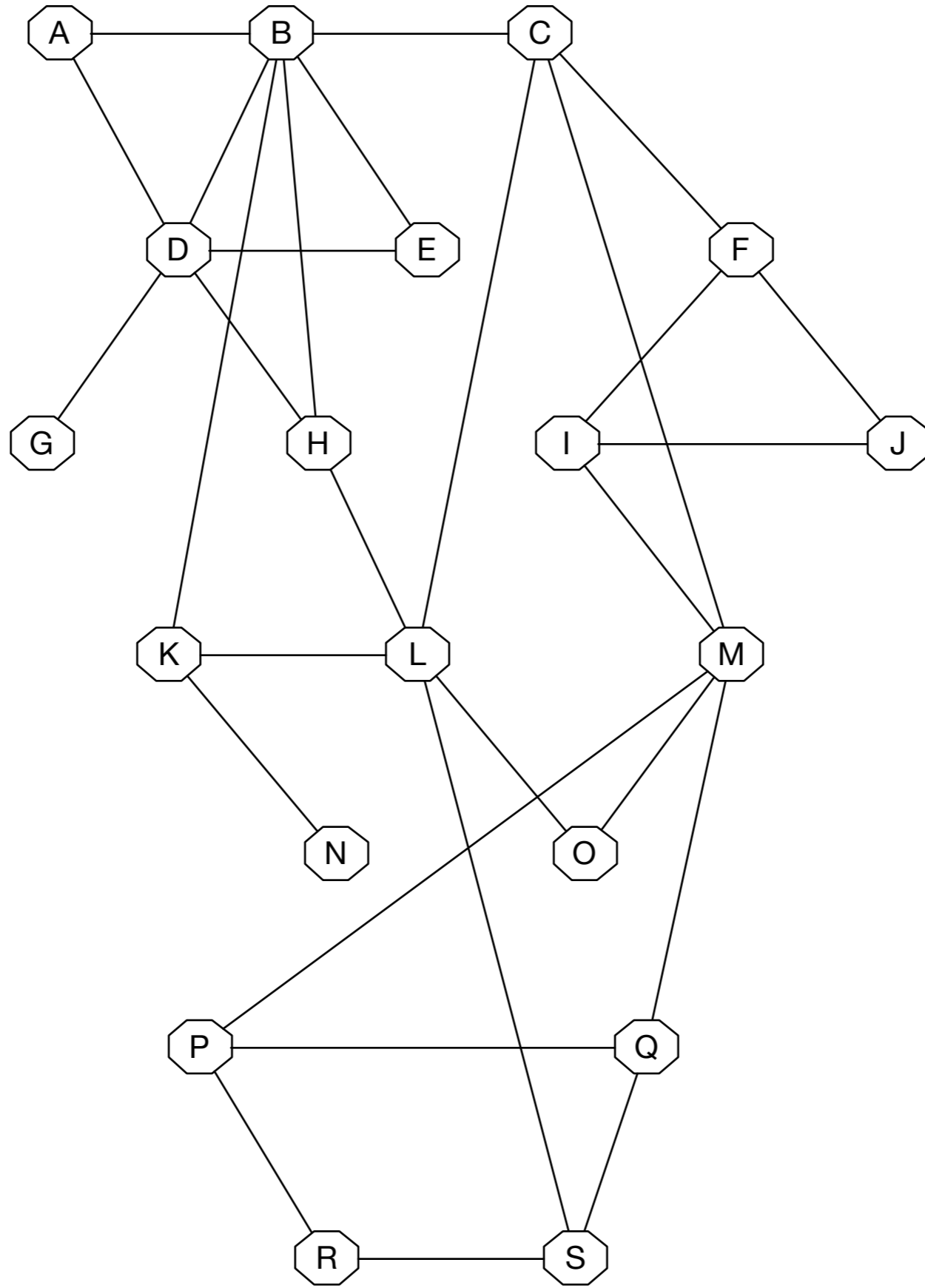
- Example: Calculate shortest path from A to D



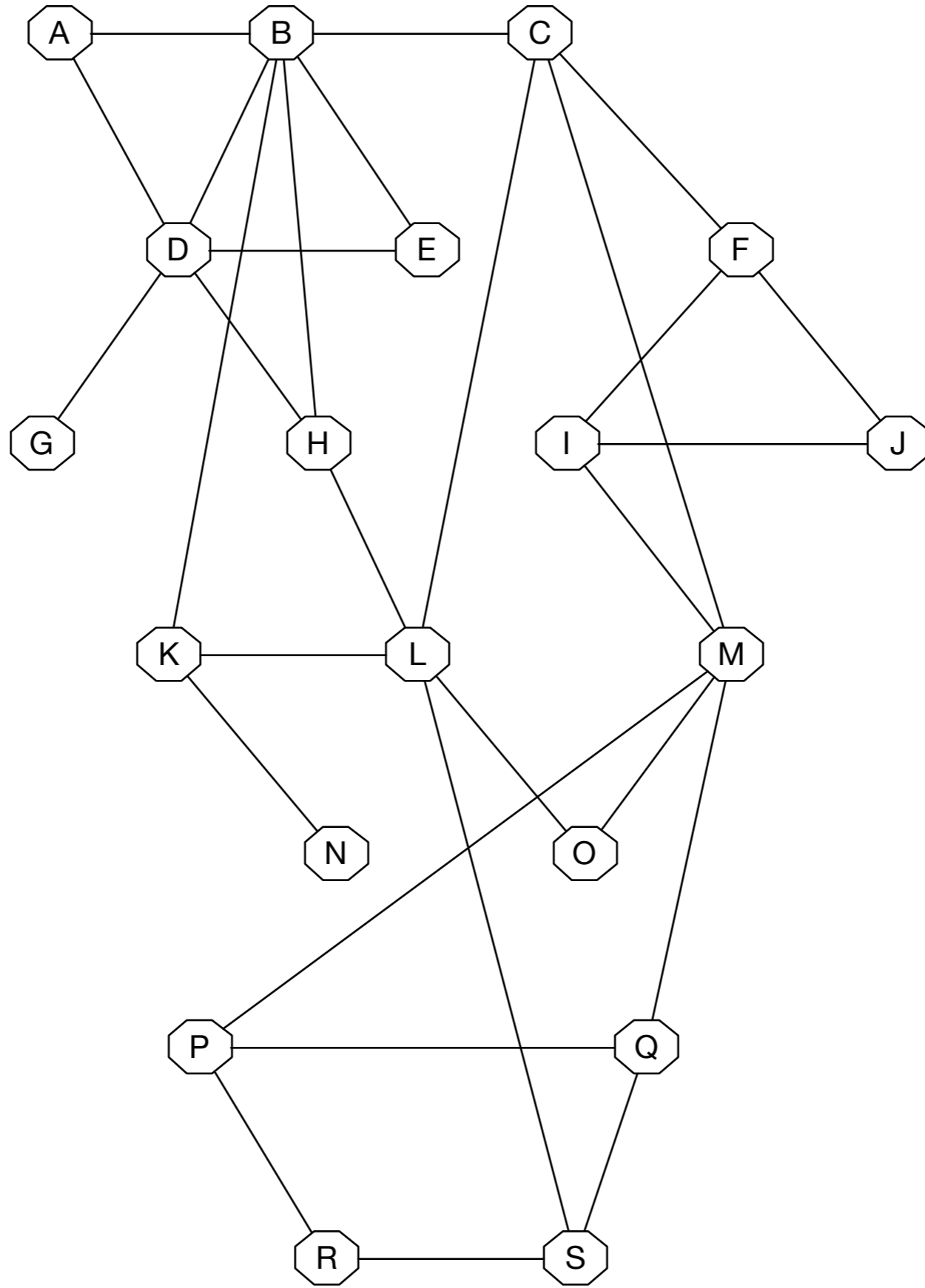
# Shortest Path Algorithm

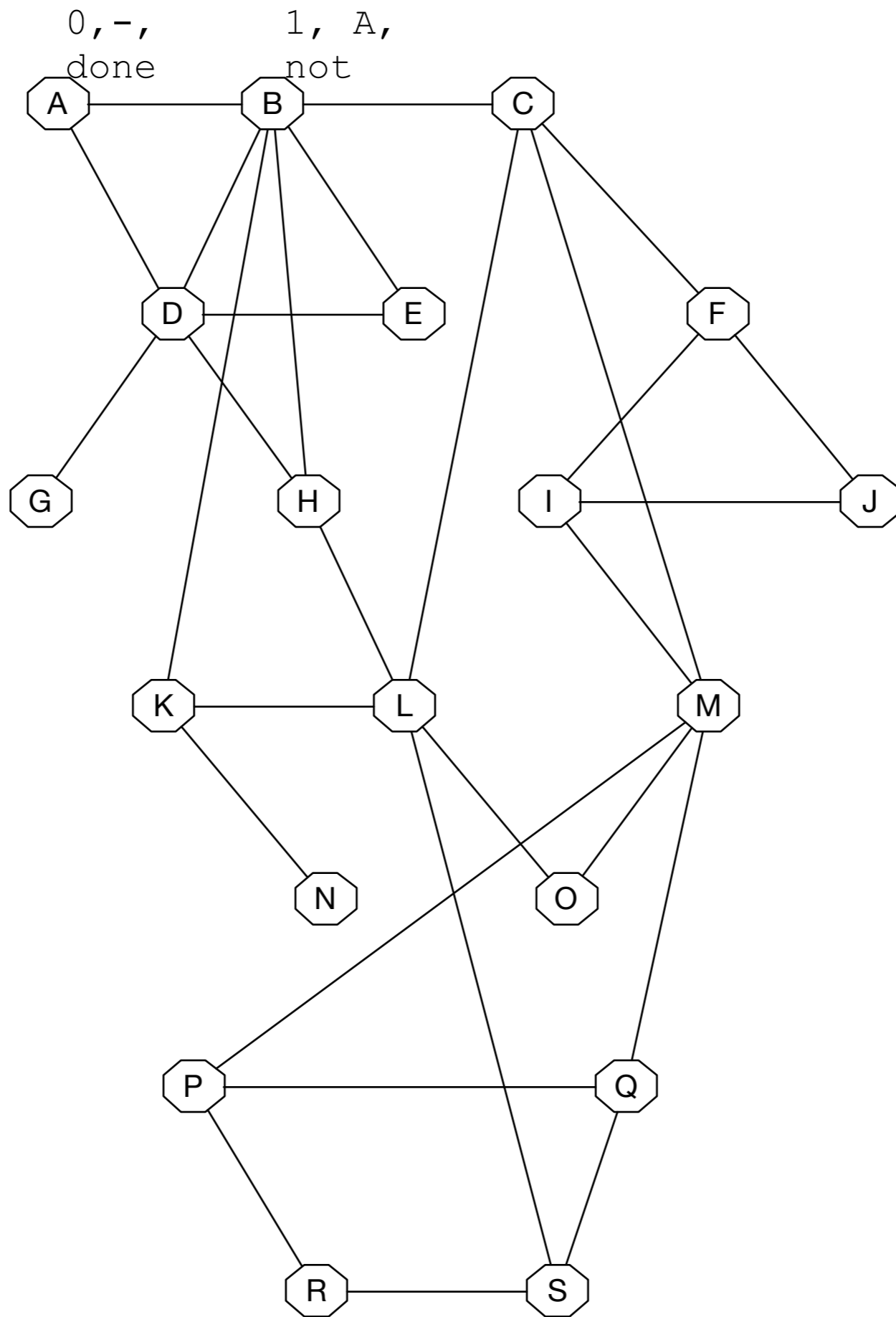
```
for (p = &state[0]; p < &state[n]; p++) {      /* initialize state */
    p->predecessor = -1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0; state[t].label = permanent;
k = t;                                          /* k is the initial working node */
do {                                          /* Is there a better path from k? */
    for (i = 0; i < n; i++)                 /* this graph has n nodes */
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }
}

k = 0; min = INFINITY;
for (i = 0; i < n; i++)
    if (state[i].label == tentative && state[i].length < min) {
        min = state[i].length;
        k = i;
    }
state[k].label = permanent;
} while (k != s);
```

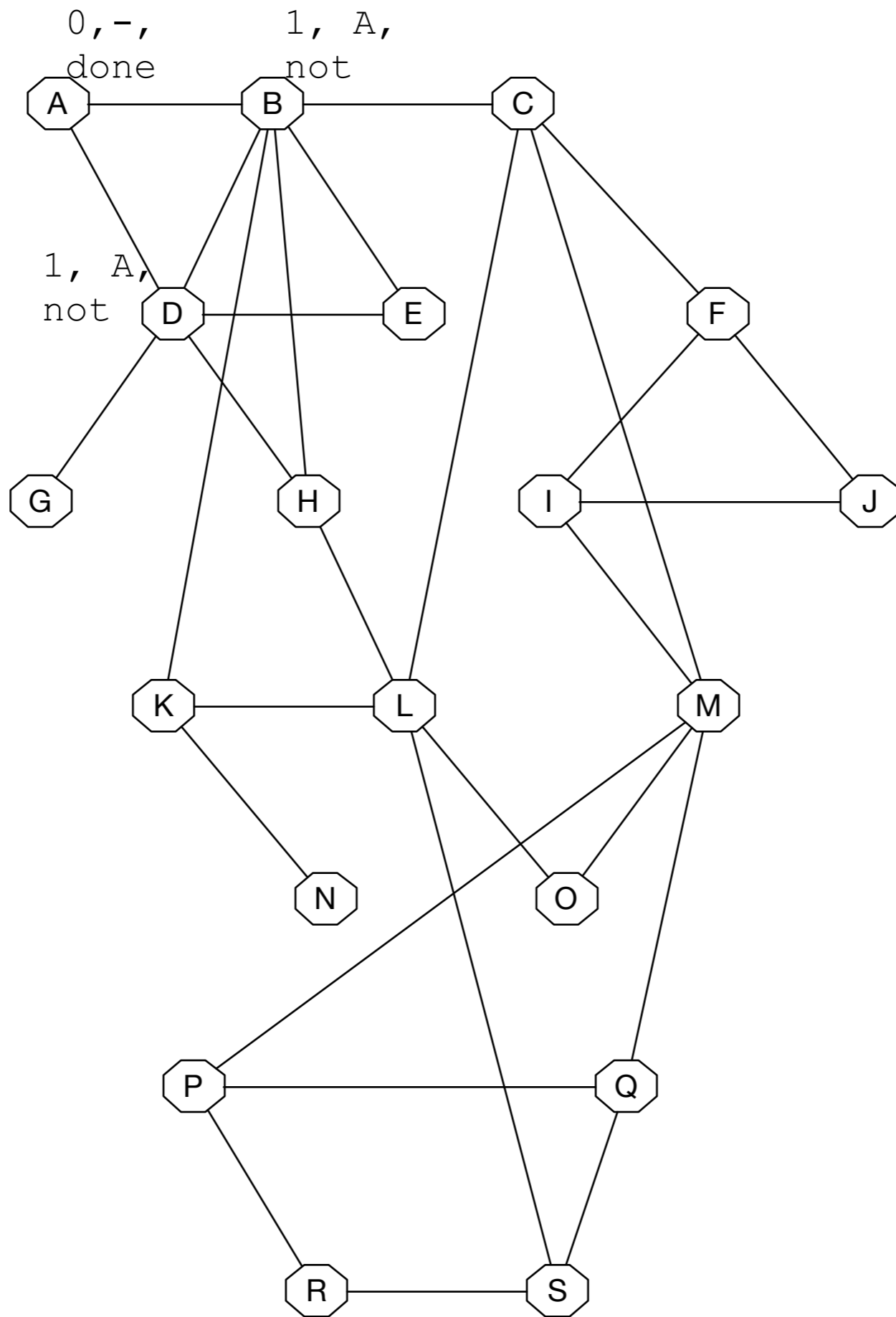


0, -, done

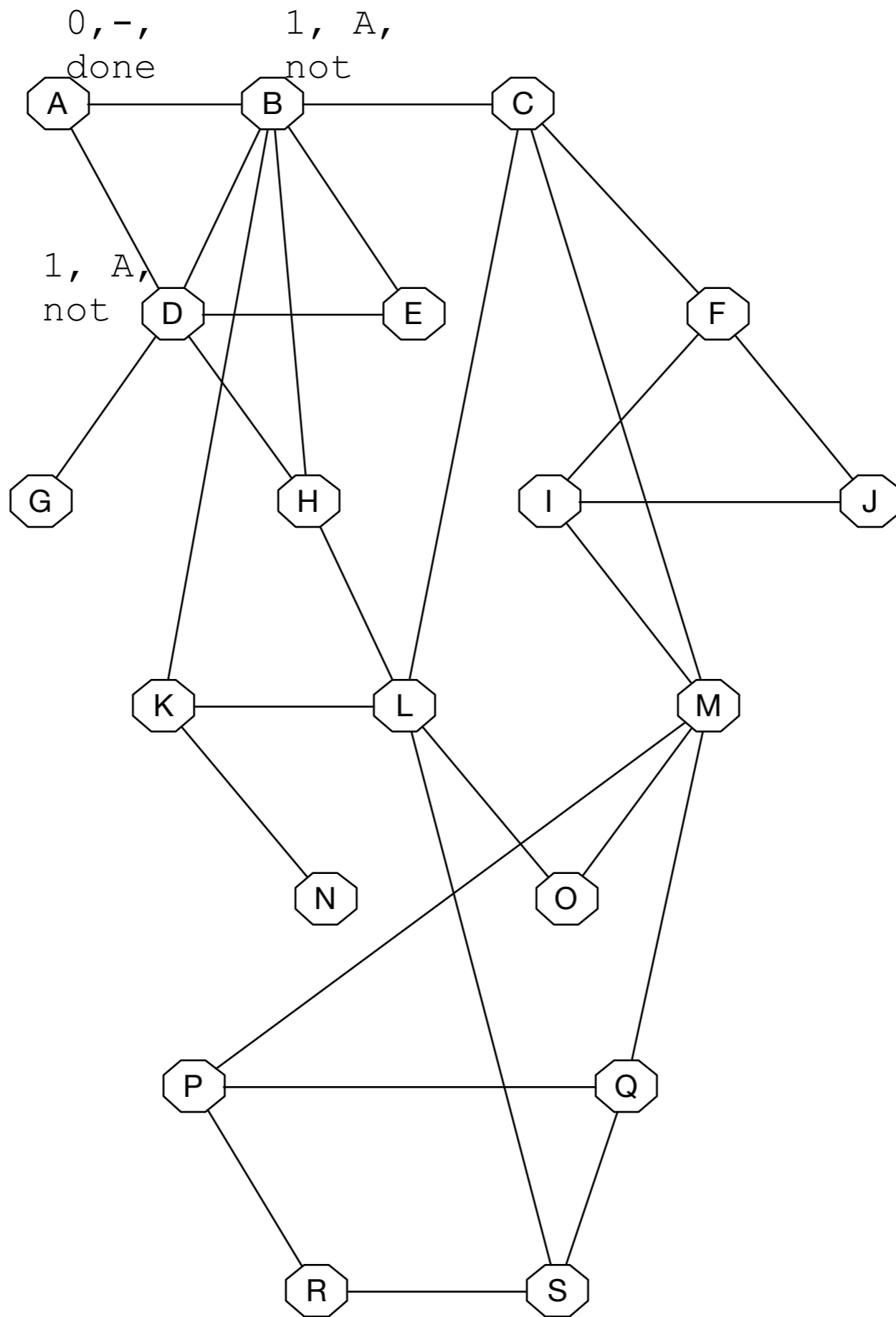




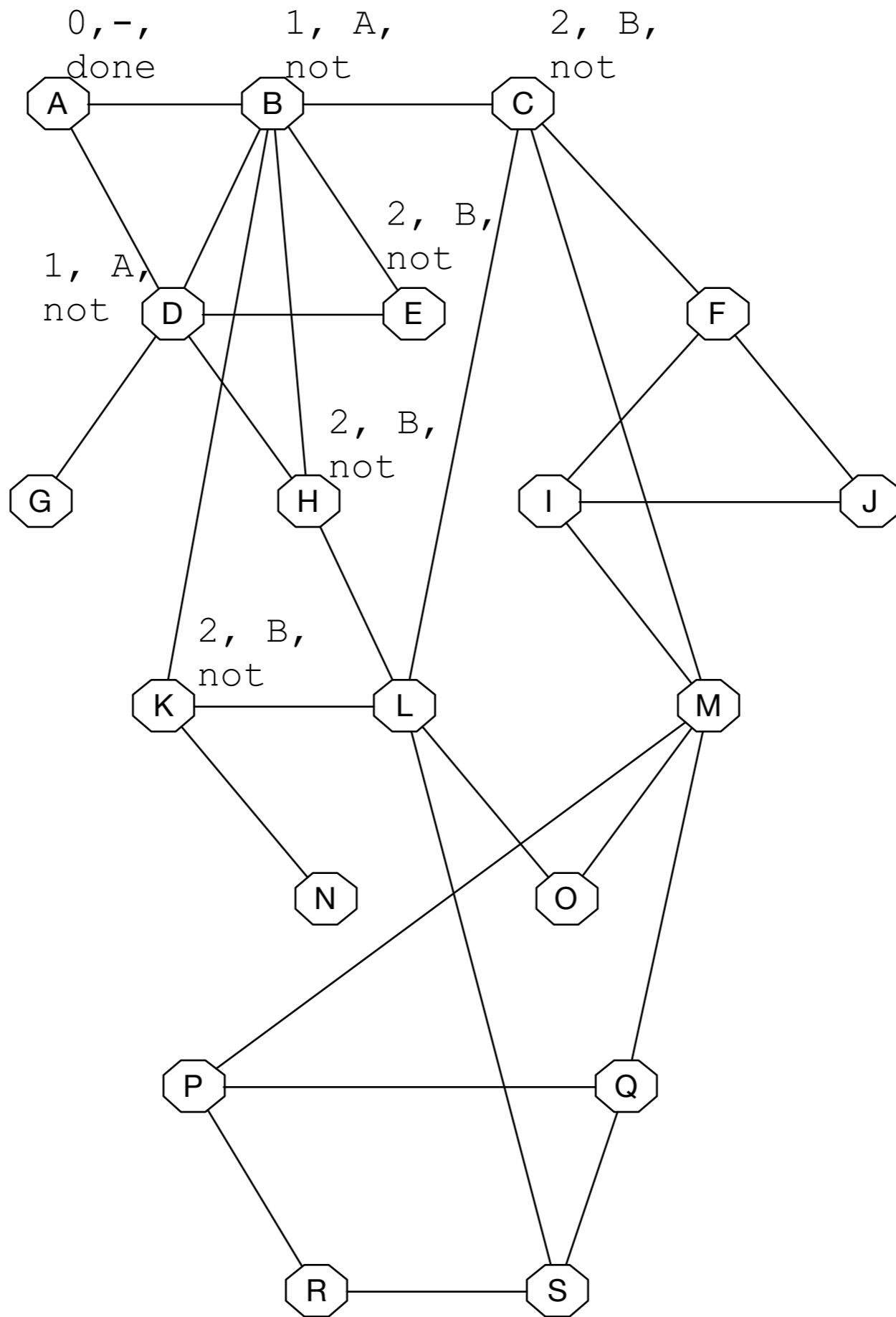




Queue:  
A

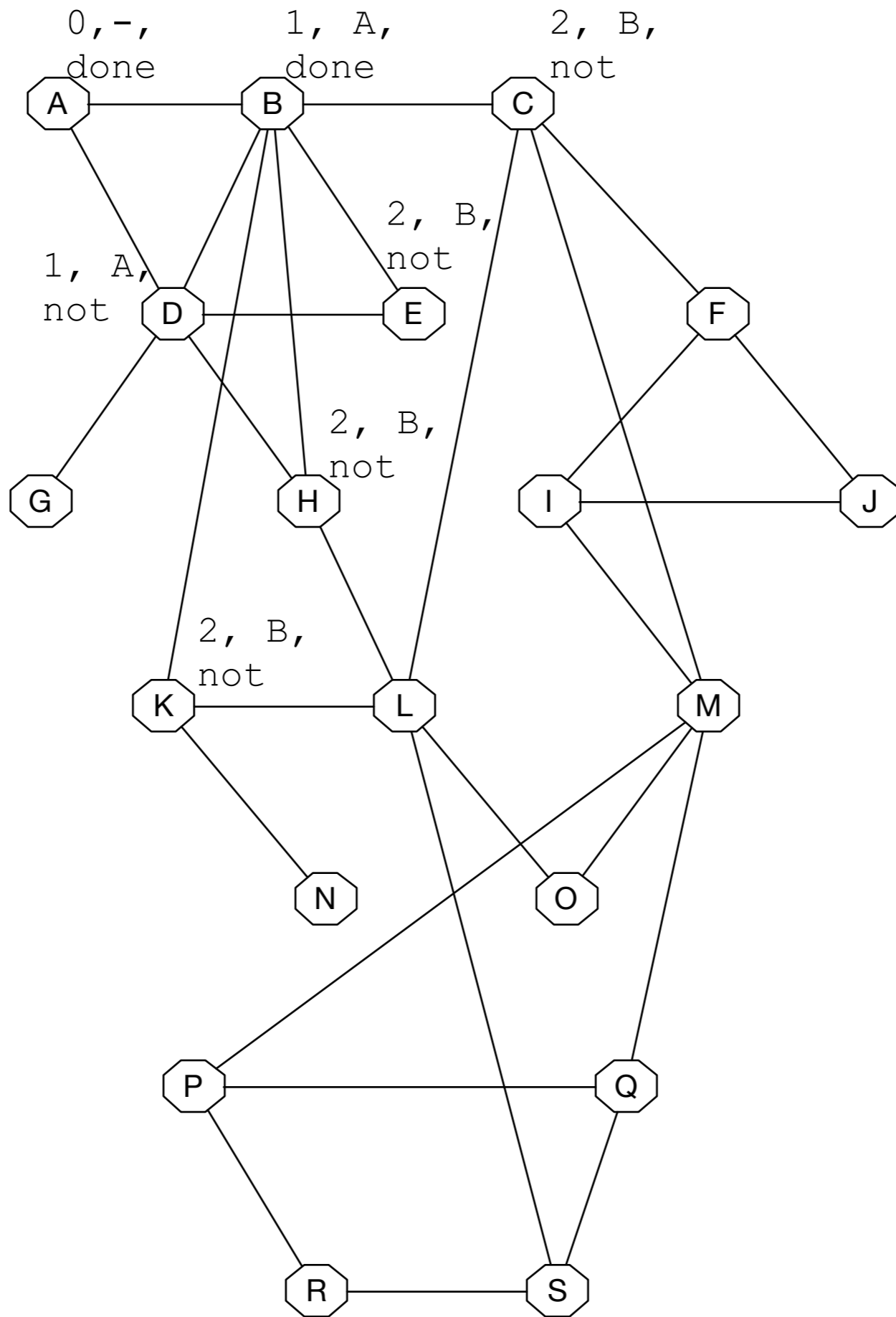


Queue:  
B  
D

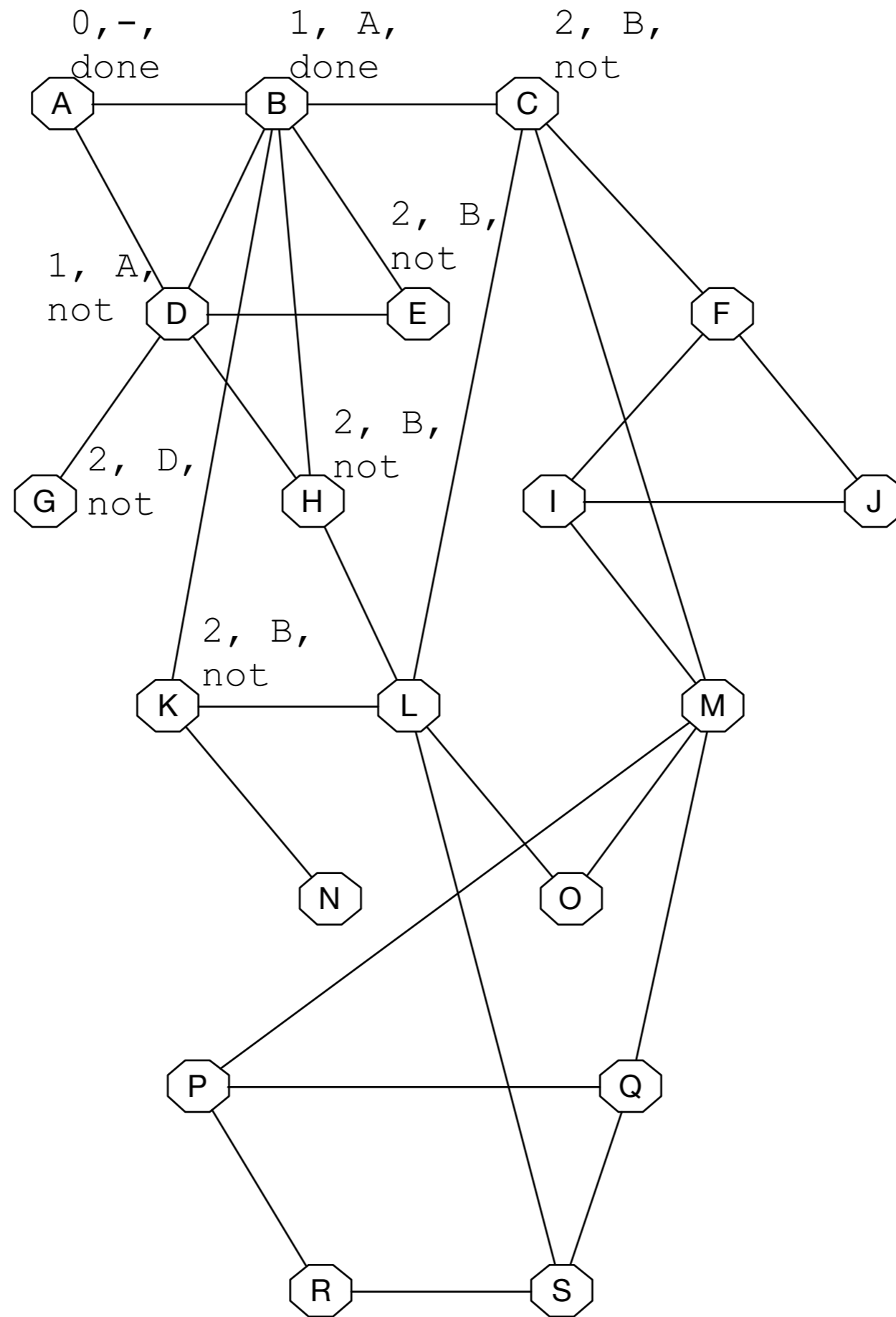


Queue:

B  
D  
C  
E  
H  
K

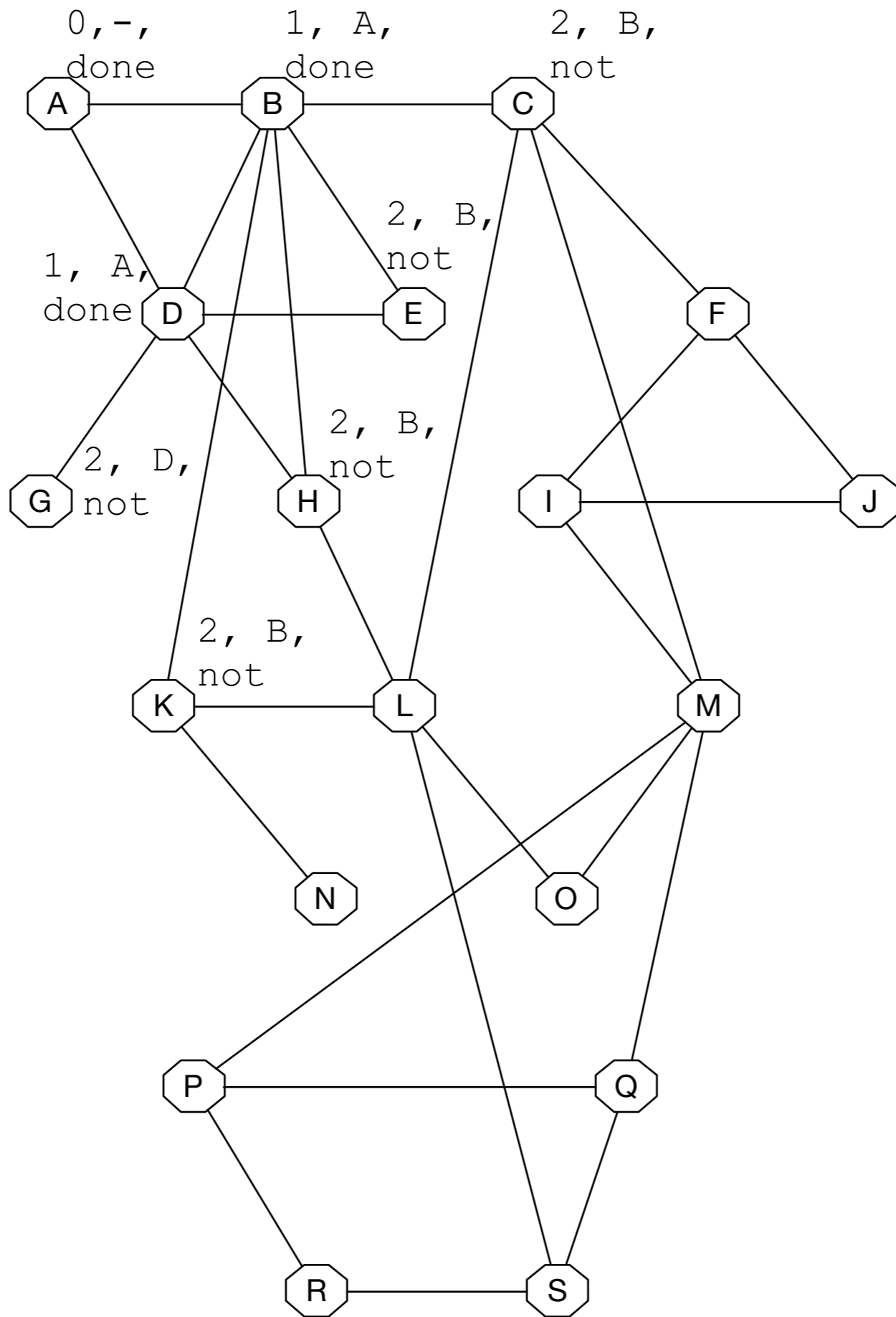


Queue:  
D  
C  
E  
H  
K

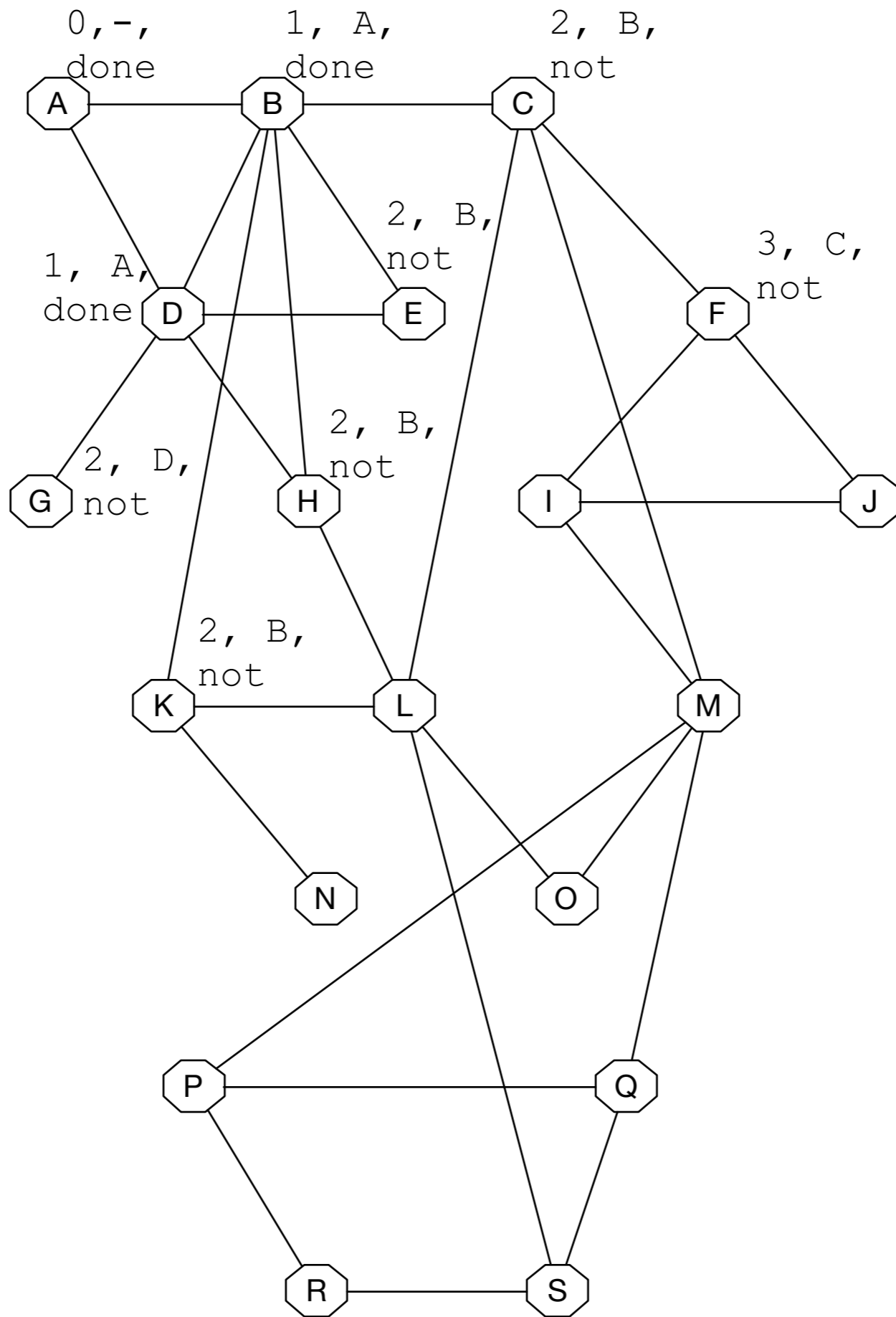


Queue:

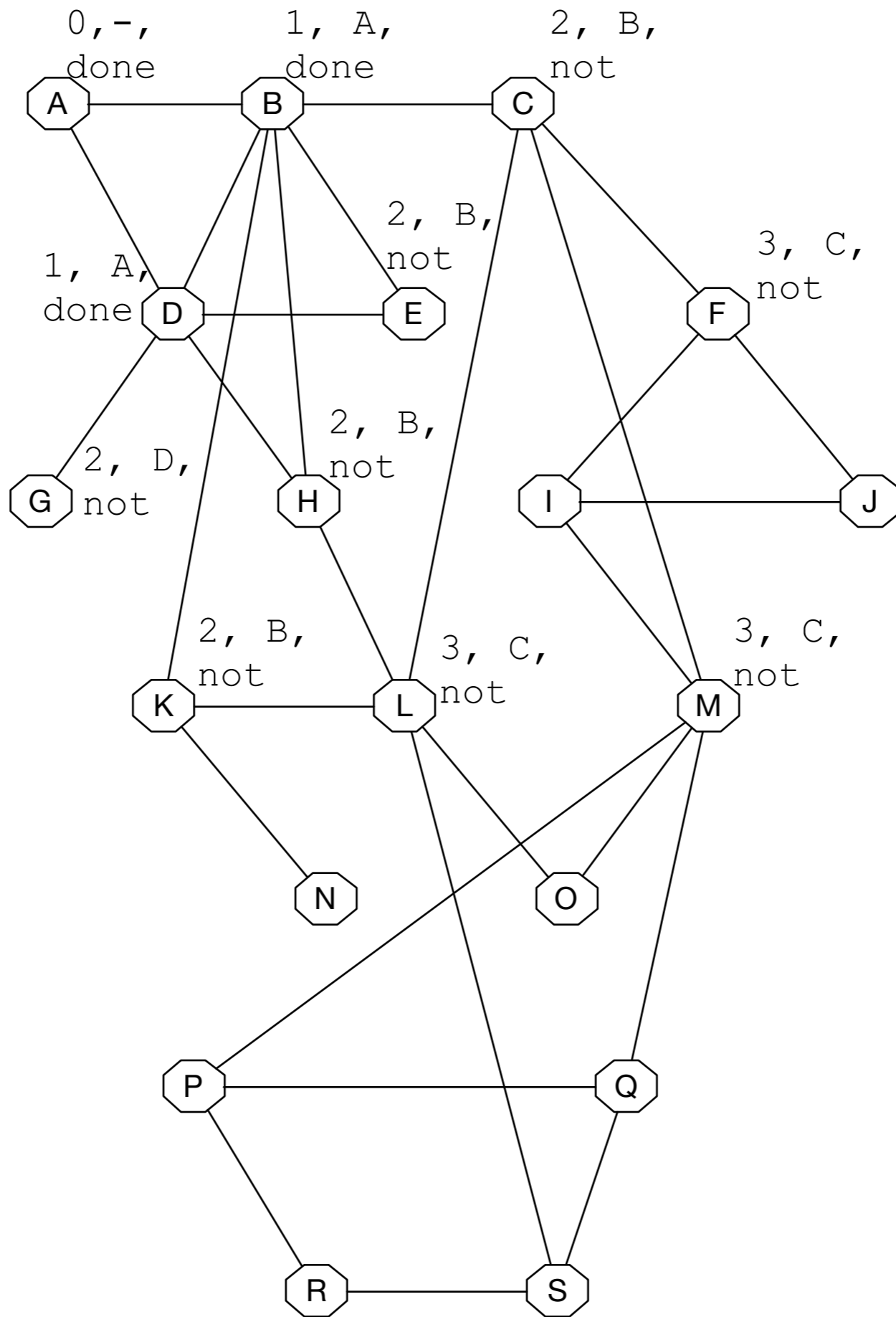
D  
C  
E  
H  
K  
G



Queue:  
C  
E  
H  
K  
G



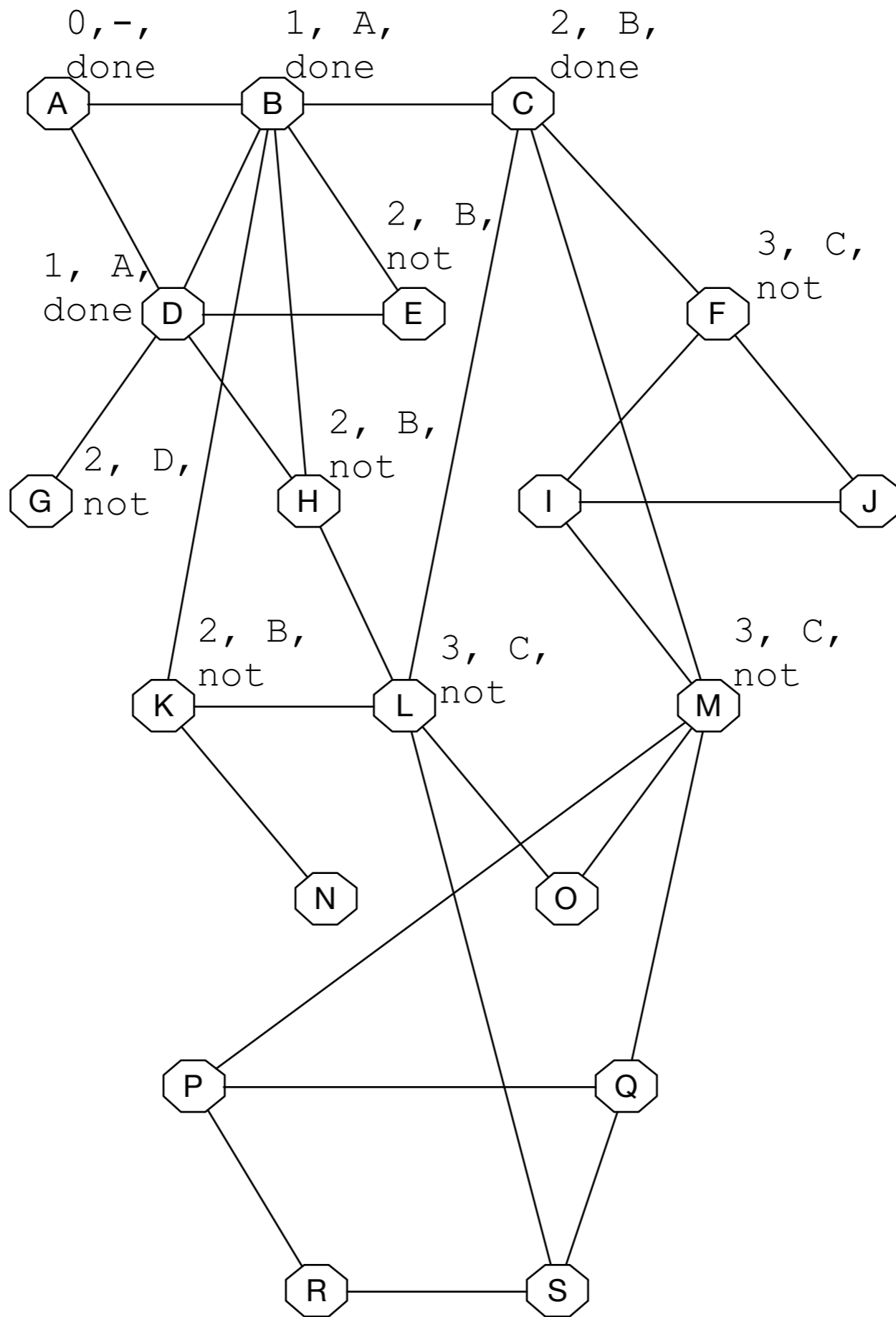
Queue:  
 C  
 E  
 H  
 K  
 G  
 F



Queue:

C  
E  
H  
K  
G  
F  
L  
M





Queue:

E  
H  
K  
G  
F  
L  
M

# Shortest Path Algorithm

- Dijkstra's algorithm
  - defaults to Breadth First Search if all edge costs are 1
  - is centralized
    - Source needs to get information about the complete network

# Routing Algorithms

- Routing algorithms
  - adaptive vs. non-adaptive
    - Can we react to changes in the network?
- Example: Routing between ISP
  - Service agreement (peering relationship) between ISP
    - **Hot Potato Routing:**
      - ISP hands over packet asap to other ISP
    - **Cold Potato Routing:**
      - ISP keeps packet as much as possible inside own network

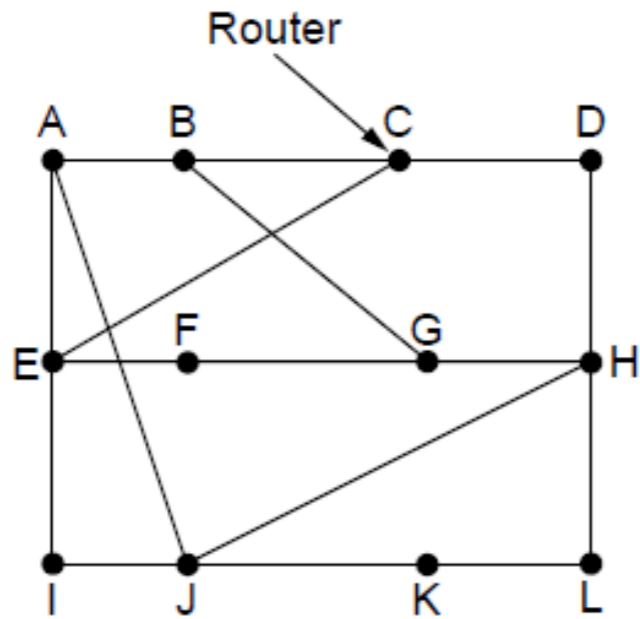
# Flooding

- A simple routing method to send a packet to all network nodes
- Each node floods a new packet received on an incoming link by sending it out all of the other links
- Nodes need to keep track of flooded packets to stop the flood; even using a hop limit can blow up exponentially

# Distance Vector Routing

- Distance vector is a distributed routing algorithm
  - Shortest path computation is split across nodes
- Algorithm:
  - Each node knows distance of links to its neighbors
  - Each node advertises vector of lowest known distances to all neighbors
  - Each node uses received vectors to update its own
  - Repeat periodically

# Distance Vector Routing

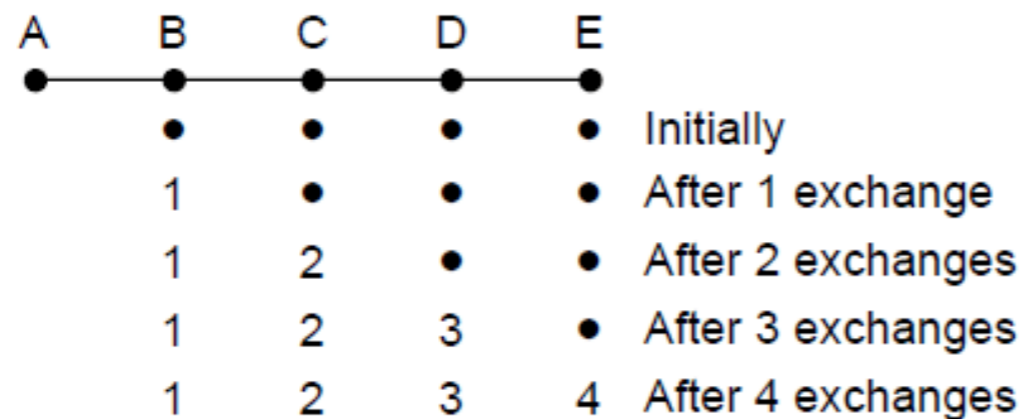


To	A	I	H	K	New estimated delay from J	
					↓ Line	
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

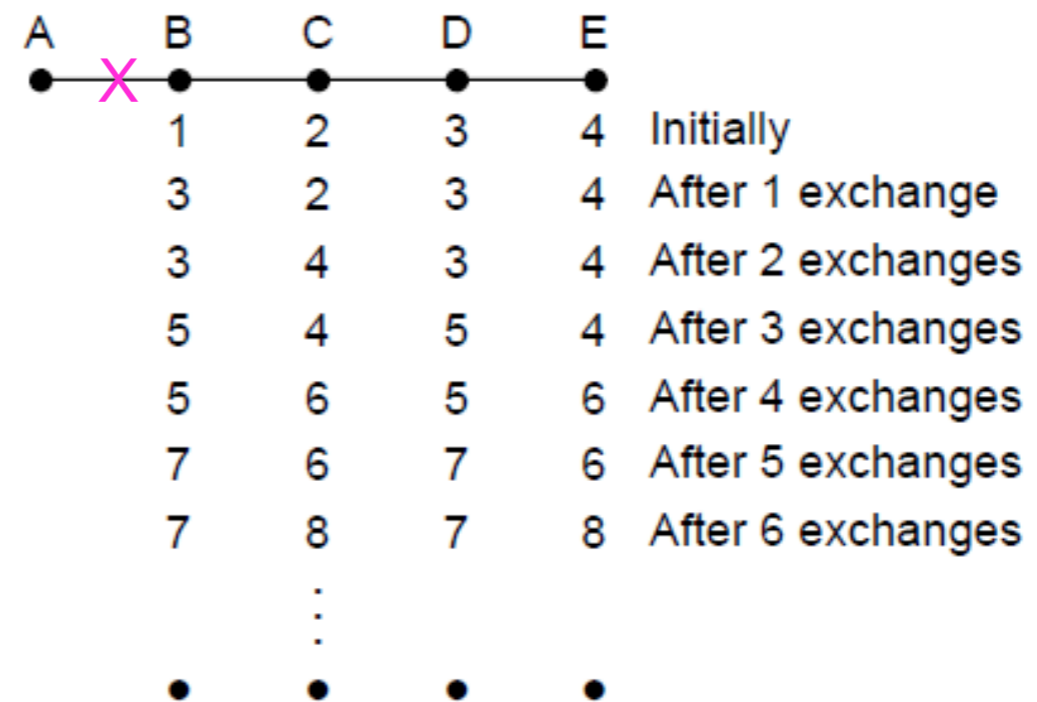
Vectors received from J's four neighbors

# The Count-to-Infinity Problem

Failures can cause DV to “count to infinity” while seeking a path to an unreachable node



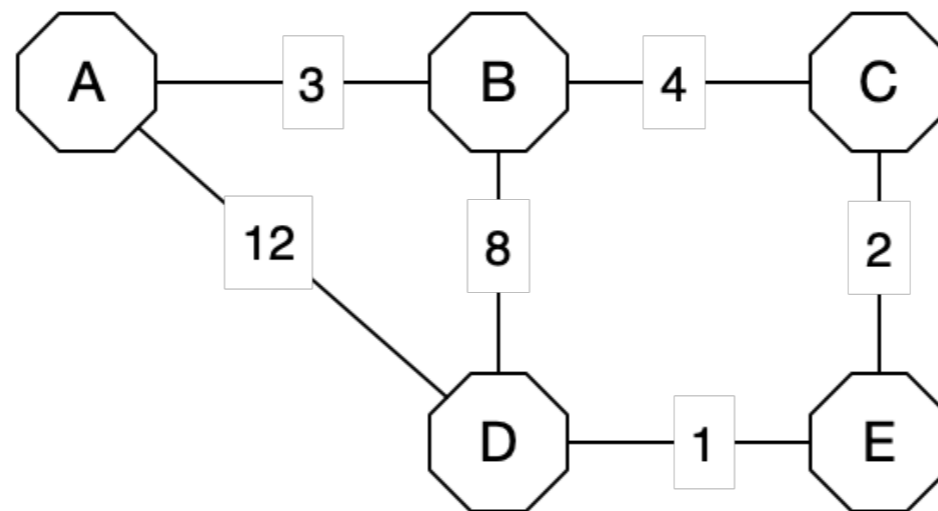
Good news of a path to A spreads quickly



Bad news of no path to A is learned slowly

# Quiz

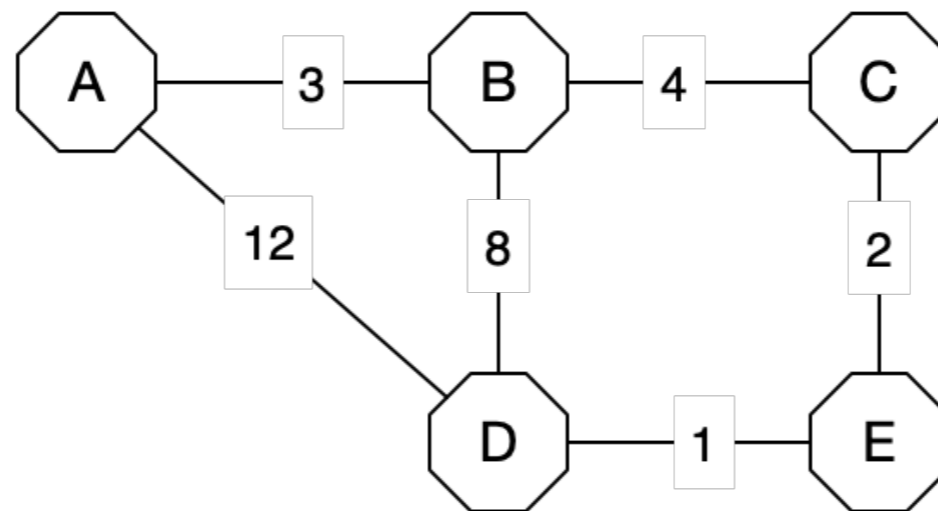
- What is the distance vector that A will send out (before receiving any other routing messages)?





# Quiz

- Assume C has received from B the distance vector (A:3, D:8, C:4) and from E the distance vector (A:13, B:9, C:2).
- What is the new table for C:

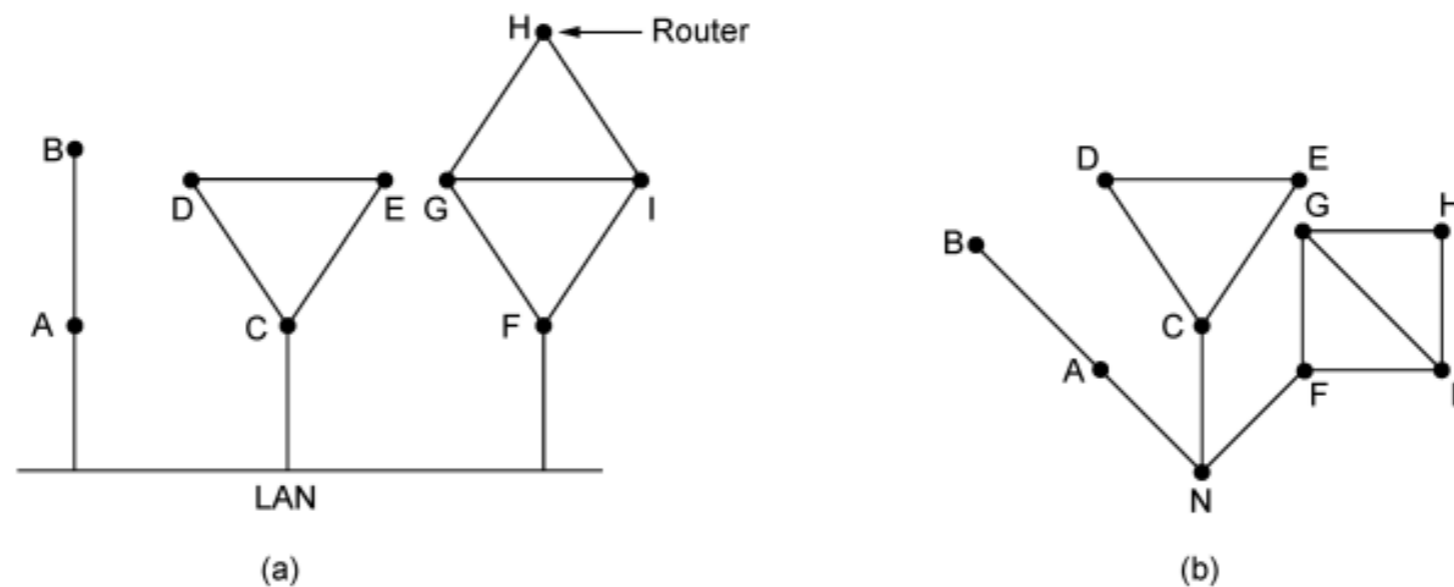


# Link State Routing

- Link state is an alternative to distance vector
  - More computation but simpler dynamics
  - Widely used in the Internet (OSPF, ISIS)
- Algorithm:
  - Each node floods information about its neighbors in LSPs (Link State Packets); all nodes learn the full network graph
  - Each node runs Dijkstra's algorithm to compute the path to take for each destination

# Link State Routing

- Learning about neighbors
  - We model the network as a graph
  - Have difficulty with a LAN with several routers
  - Solve this by creating an artificial node
  - Assign one of the routers in the LAN to act for the artificial node

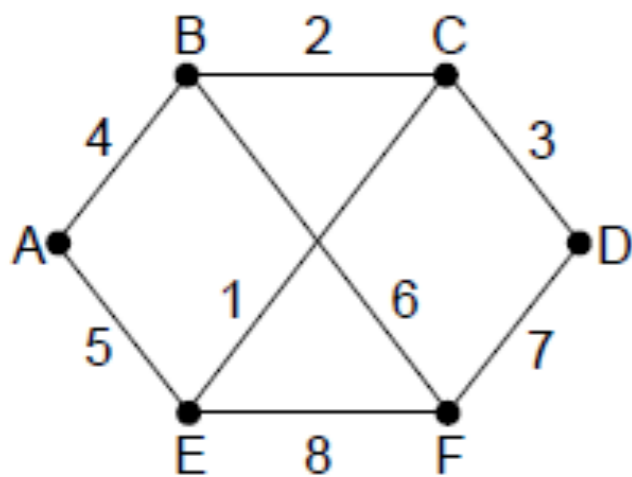


(a) Nine routers and a broadcast LAN. (b) A graph model of (a).

# Link State Routing

- Routers gather link information
  - Create link state packets

LSP (Link State Packet) for a node lists neighbors and weights of links to reach them



Network

	Link	State		Packets	
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B   4	A   4	B   2	C   3	A   5	B   6
E   5	C   2	D   3	F   7	C   1	D   7
	F   6	E   1		F   8	E   8

LSP for each node

# Link State Routing

## Reliable Flooding

- Use sequence number and age for reliable flooding
  - New LSP are acknowledged on the lines they are received and forwarded on all other lines
- Example shows the LSP database at router B

Source	Seq.	Age	Send flags			ACK flags			Data
			A	C	F	A	C	F	
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

In the example, B has links to A, C and F in the network. It received LSPs from A, C, and F directly and so acknowledged A, C, and F respectively and sent that LSP on both other links. But B received E and D on two links, so it acknowledged both and sent only on the third link.

# Link State Routing

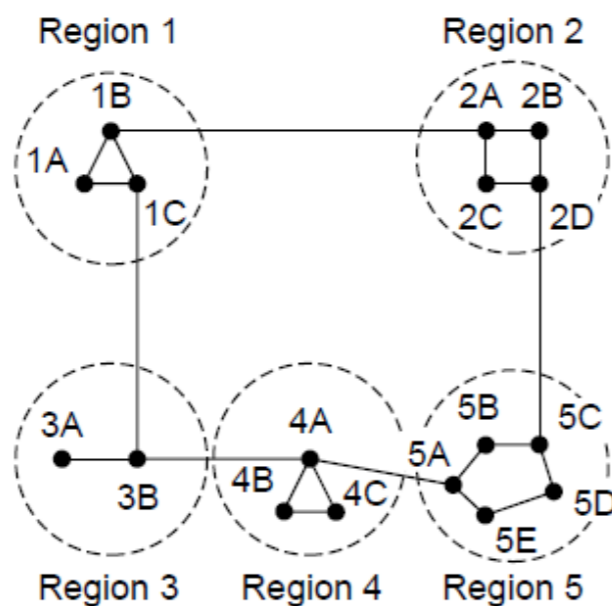
- Computing routing tables
  - If a router has a full set of link state packets
    - It knows the network
      - Every link is represented twice
        - but this is fine because the costs can be different
  - Router calculates best route to all nodes

# Hierarchical Routing

- Routing tables need to be compacted
  - Can divide nodes into regions
  - Introduces a hierarchy:
    - Routers know how to route to local nodes
    - Routers know the gateways to all other nodes
  - Hierarchy compact routing tables but path length can become bigger
- Very large networks need more than one hierarchy
- Kamoun and Kleinrock (1979):
  - $N$  router network needs  $\ln(N)$  levels with a total of  $e \ln(N)$  entries without appreciable increase in path lengths.

# Hierarchical Routing

- Hierarchical routing reduces the work of route computation
- May result in longer paths than flat routing



Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

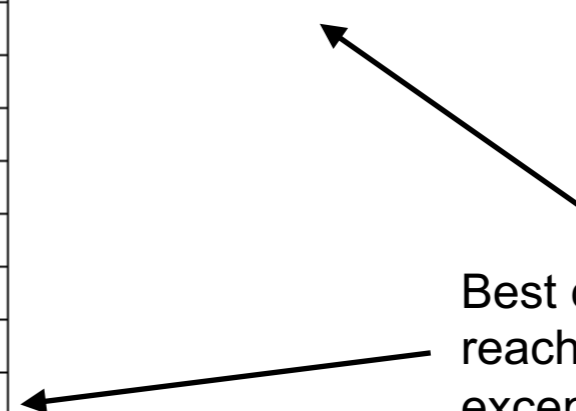
Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Hierarchical routing is what you think it is, e.g., to reach a given telephone first head towards the right country, then the right city in the country, then the phone in the city.

Each node keeps only one entry per region for other regions, plus an entry for all nodes in the local region.

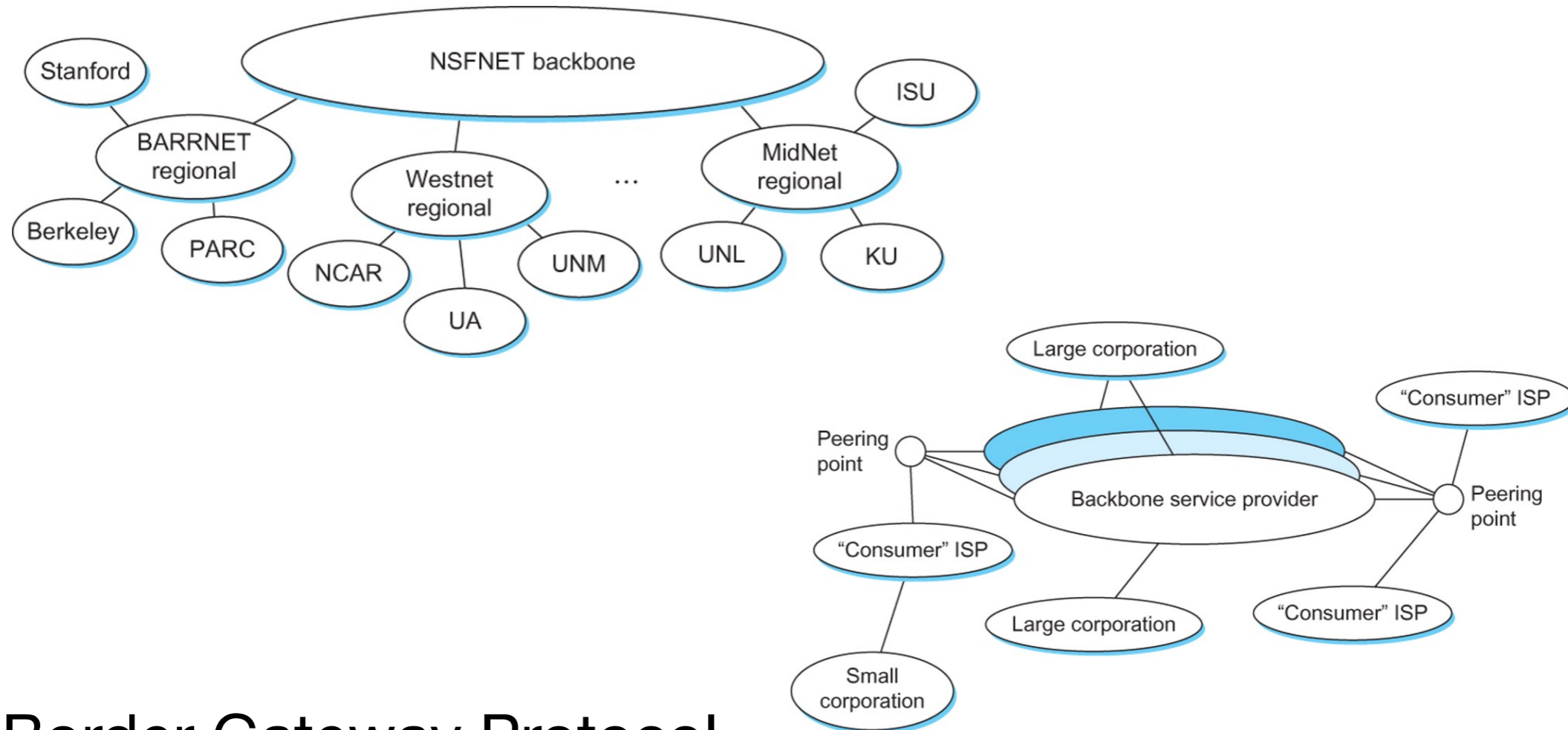
The advantages are smaller routing tables, smaller routing computations to run at nodes, and fewer/smaller messages to send to describe the network.

Best choice to reach nodes in 5 except for 5C





# Hierarchical Routing



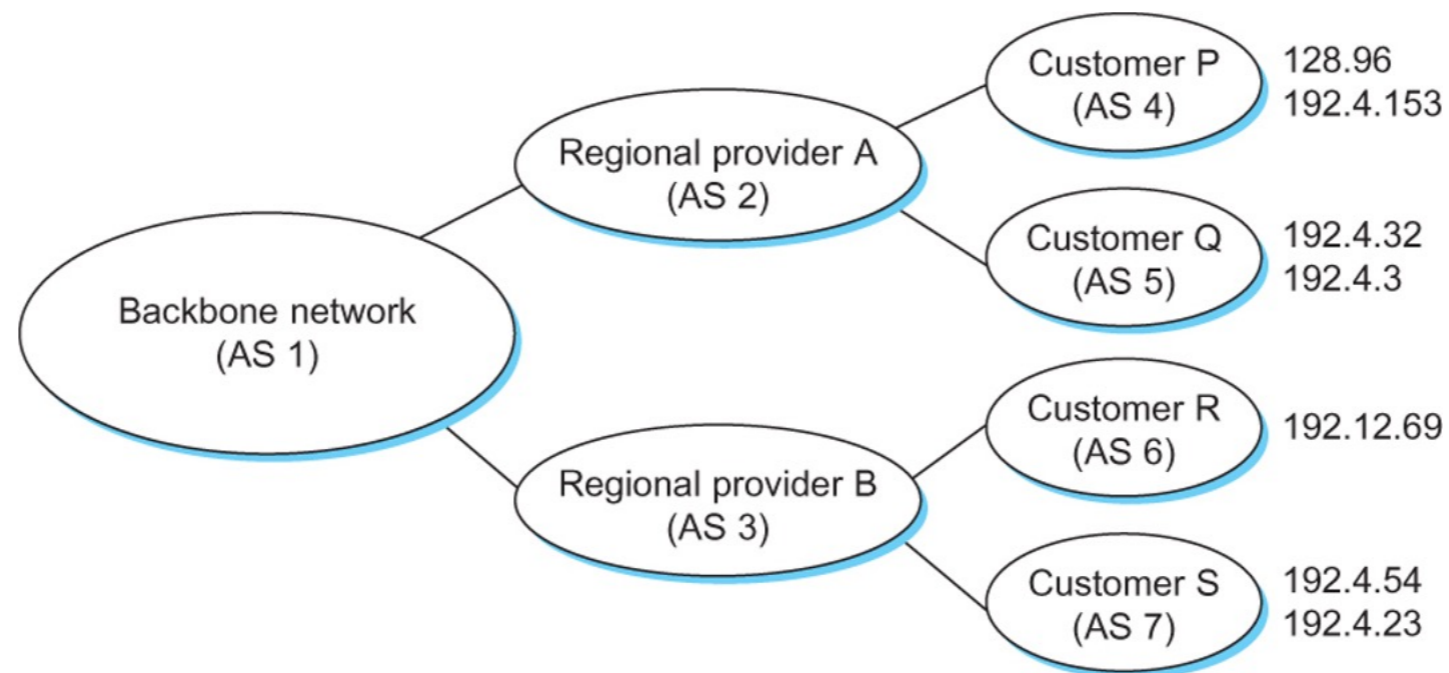
- Border Gateway Protocol
  - How to connect Autonomous Systems

# Hierarchical Routing

- Each AS has a block (or a few blocks) of IP addresses
  - Routing inside an AS is done by AS
  - Routing between AS is done by an internet-wide standard
    - Border Gateway Protocol 4

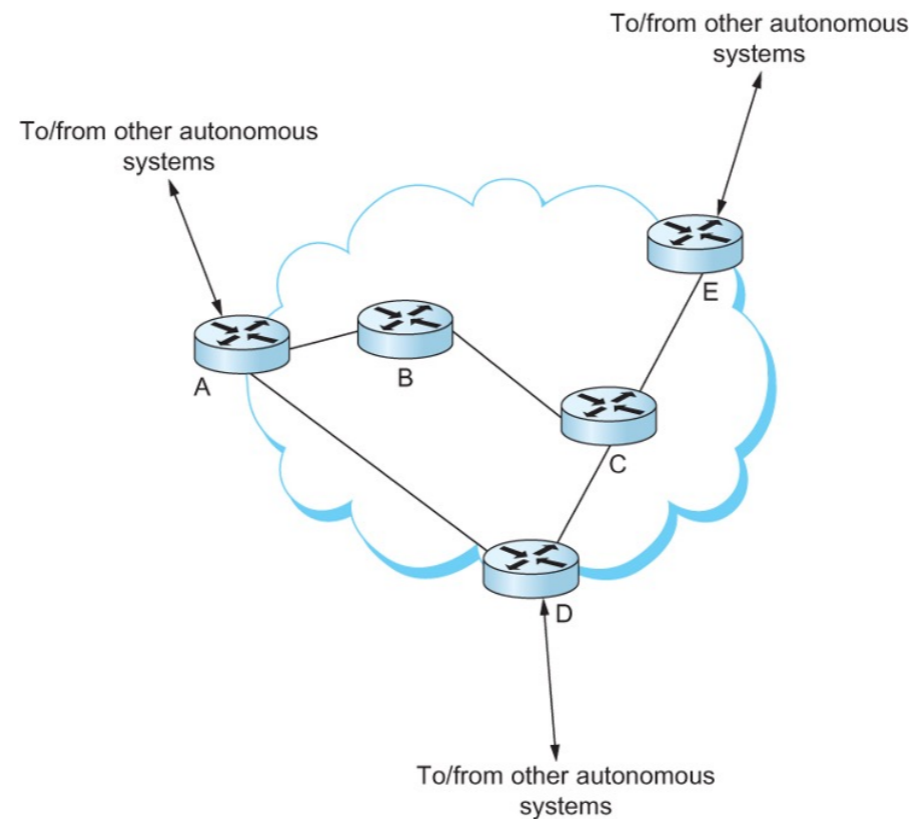
# Hierarchical Routing

- Simple Example



- Speaker for AS 2 advertises reachability to P and Q
- Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.
- Speaker for backbone network then advertises
- Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path <AS 1, AS 2>.
- Speaker can also cancel previously advertised paths

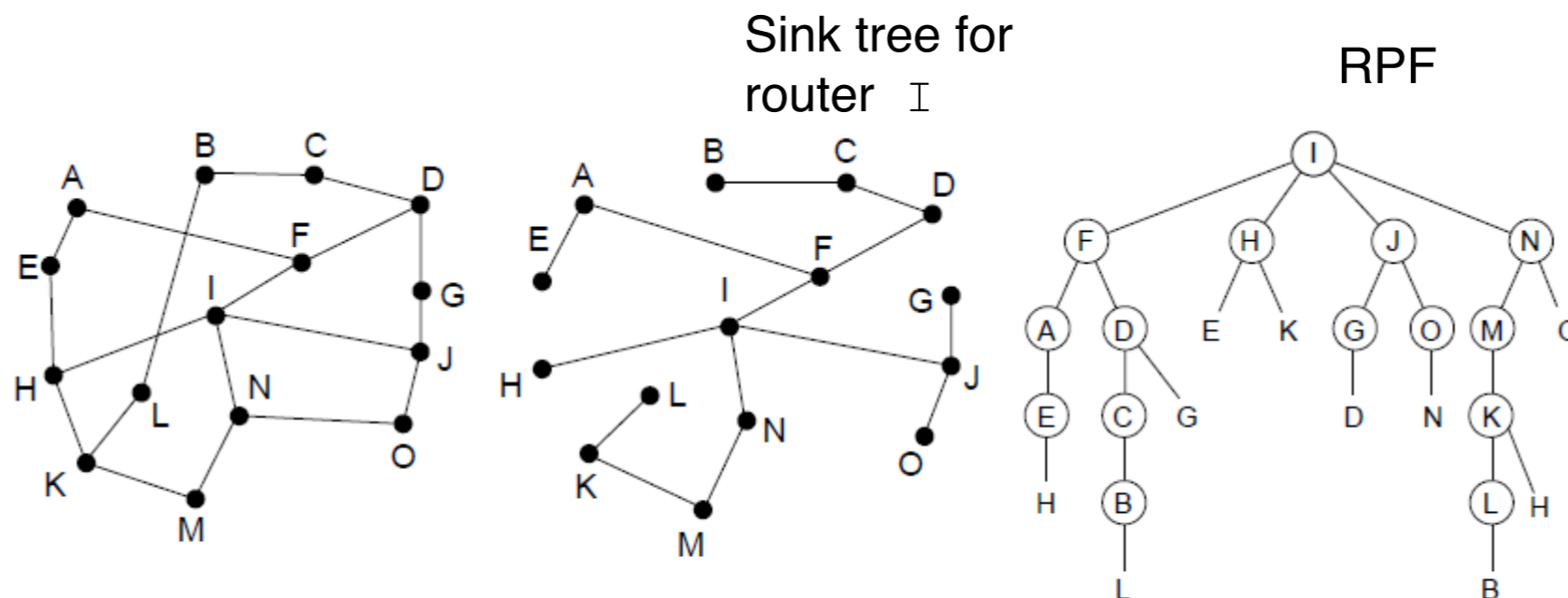
# Hierarchical Routing



- All routers run iBGP and an intradomain routing protocol.
- Border routers (A, D, E) also run eBGP to other ASs

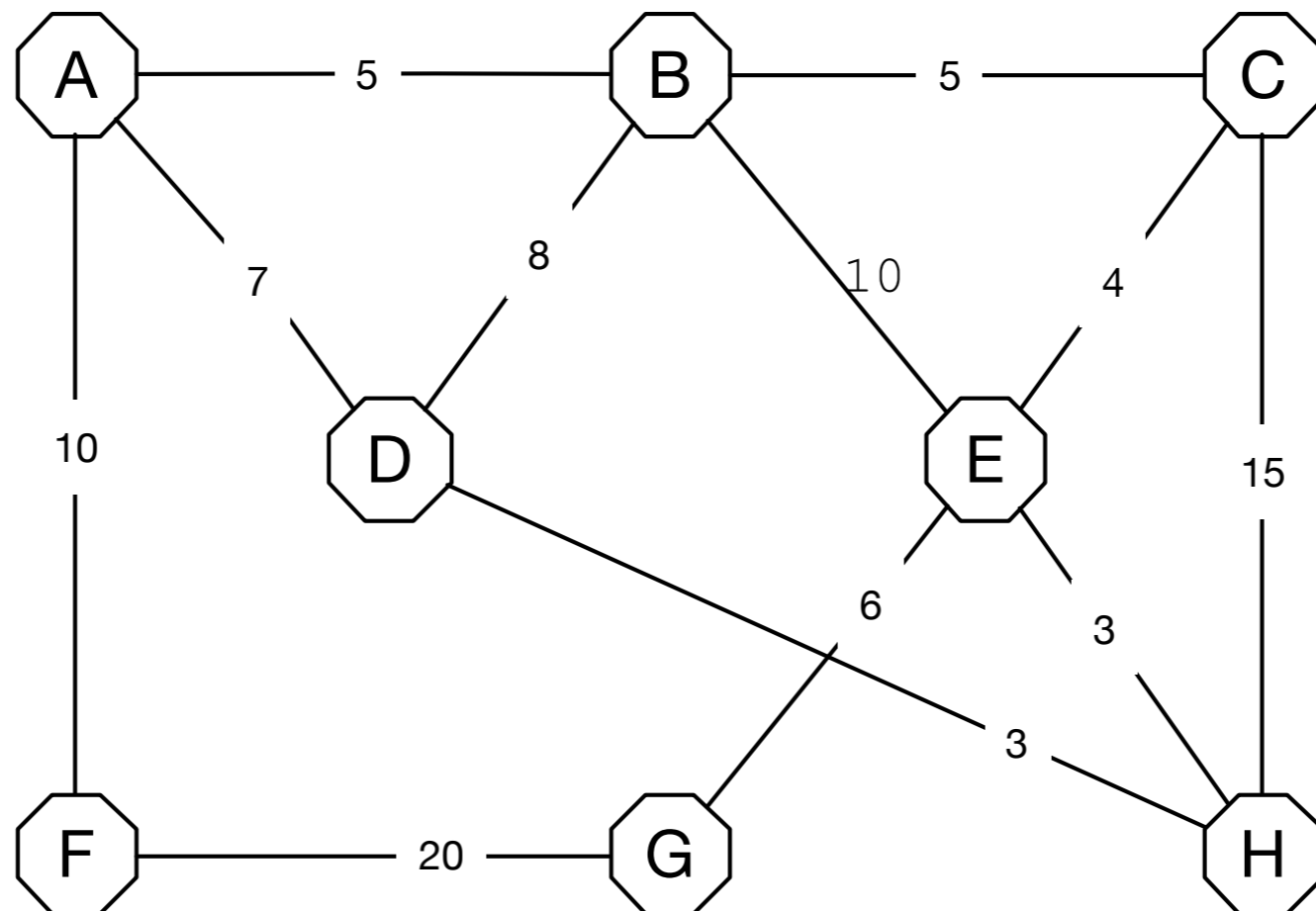
# Broadcast Routing

- Broadcast sends a packet to all nodes
  - Reverse Path Forwarding
    - If a broadcast packet arrives from the link to which we would send to the source, then it is likely to be following the best path and therefore the first copy of the broadcast.
    - We then resend it on all other links.
  - Or use sink trees at all nodes

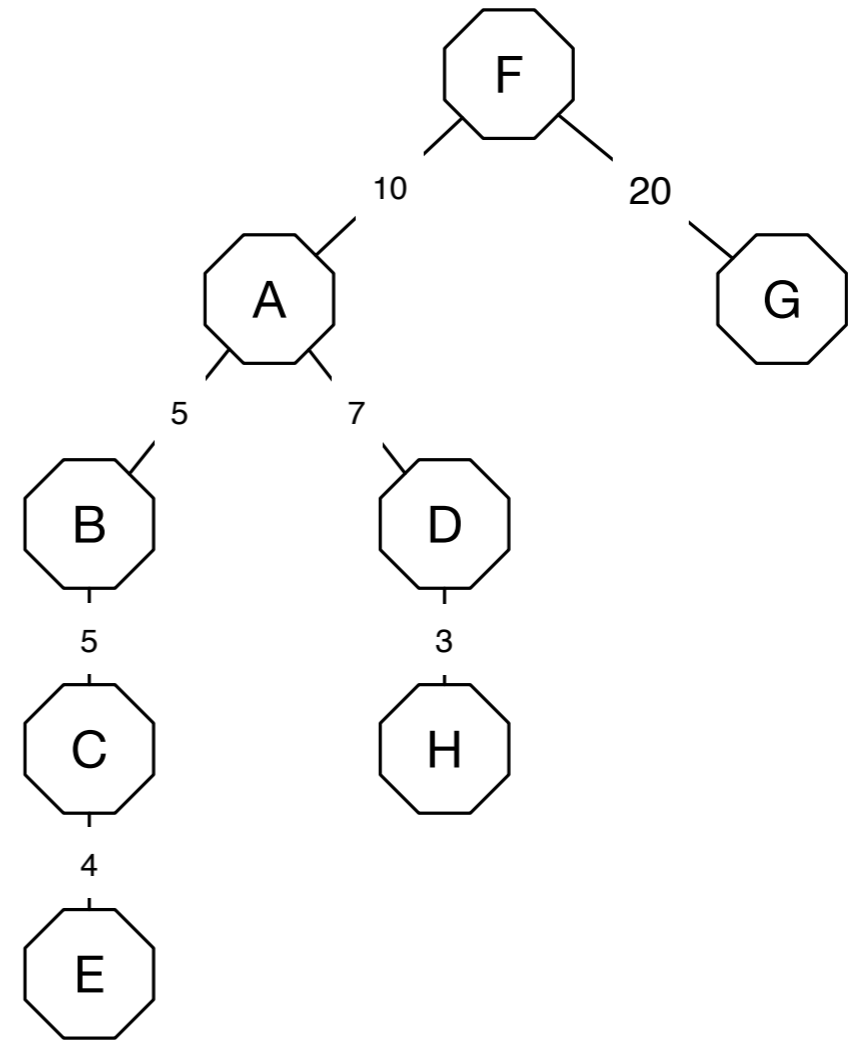
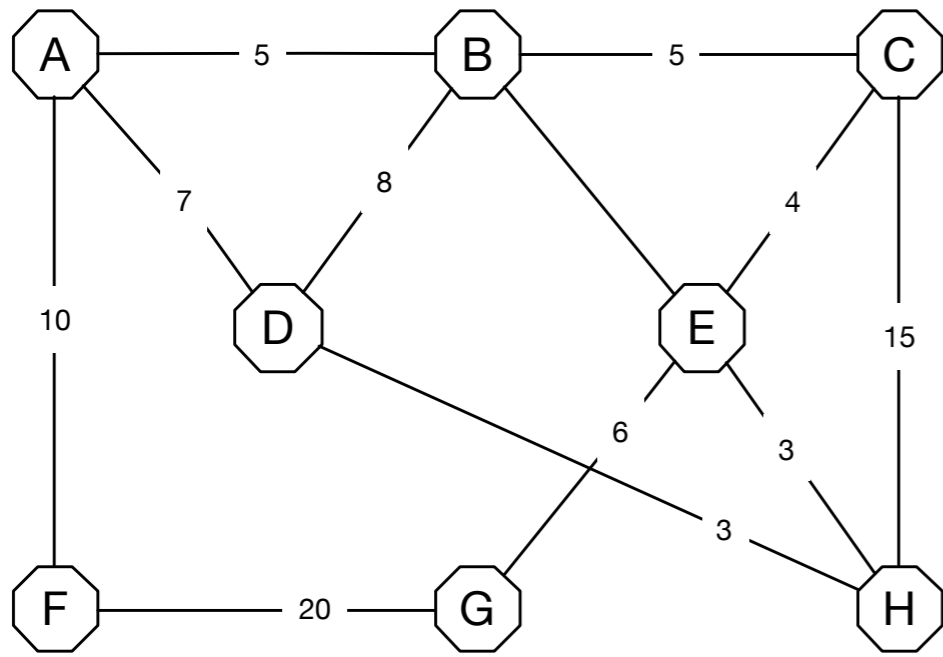


# Group Work

- Apply RPF to a broadcast message from F.



# Group Work



# Multicast Routing

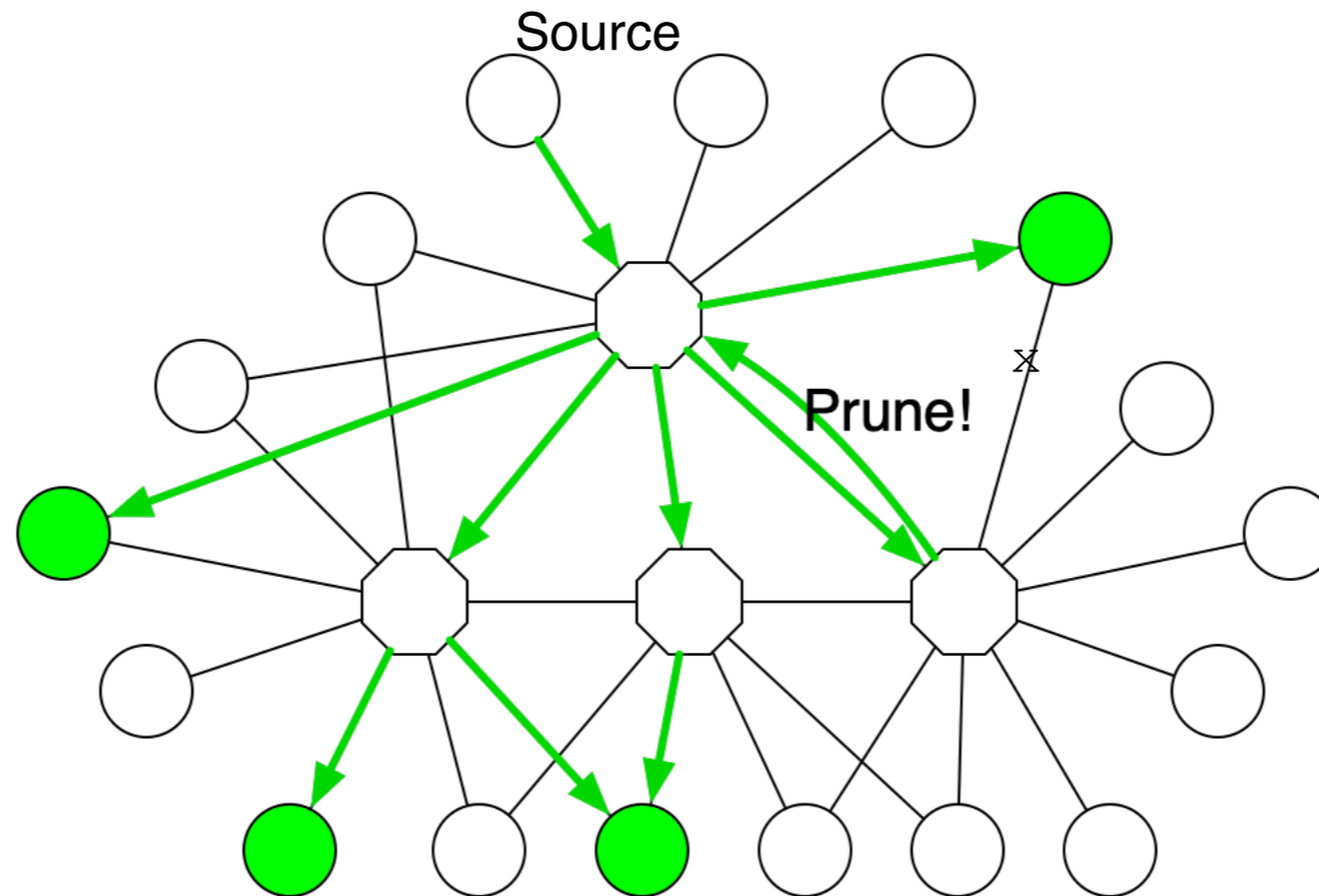
- Multicasting
  - Create (and later destroy) a group of nodes
  - Create routing entries for these groups of nodes
- Can start with a complete sink tree
  - Then prune useless links
    - Link State Protocol —> Originating router can prune itself because it knows the network
    - Distance Vector Multicast Routing Protocol (DVMRP):
      - If a router with no members of the group attached receives a multicast message, it sends a pruning message to the sending router



# Reverse Path Forwarding

- Use Prune messages to update multicast routing

- Setup:

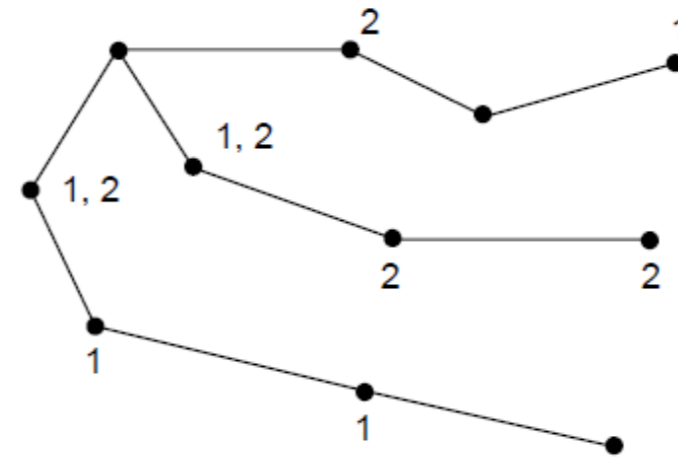
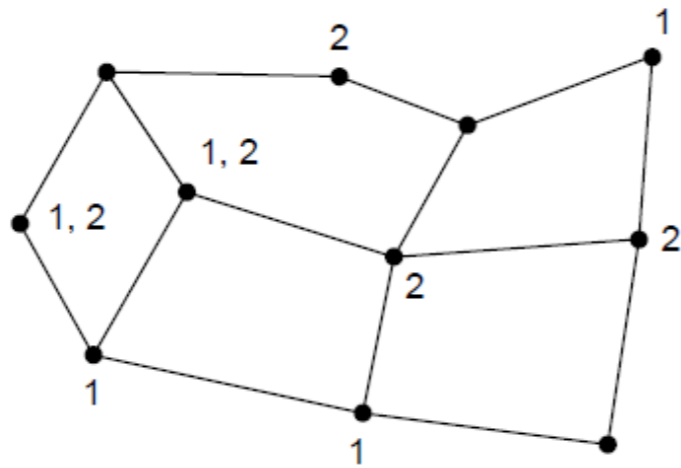


- Sending to all green nodes
- After prune message: top router only sends to two routers

# Multicast Routing

Network with two groups

Spanning tree from source

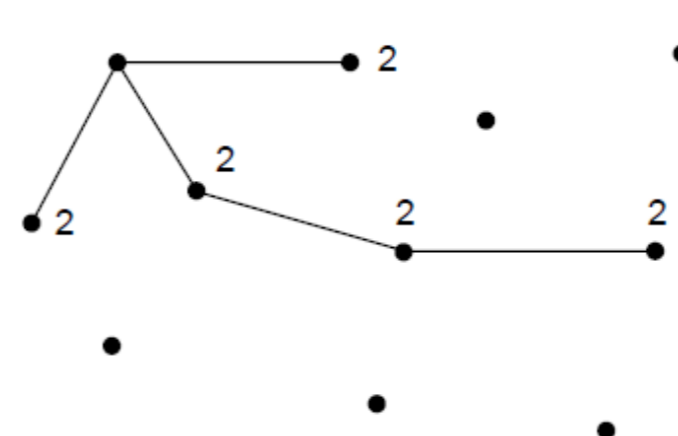
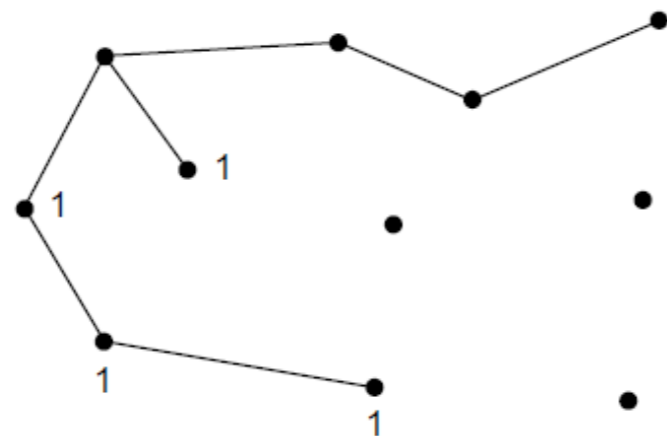


(a)

(b)

Multicast<sub>1</sub> tree for 1

Multicast tree for 2

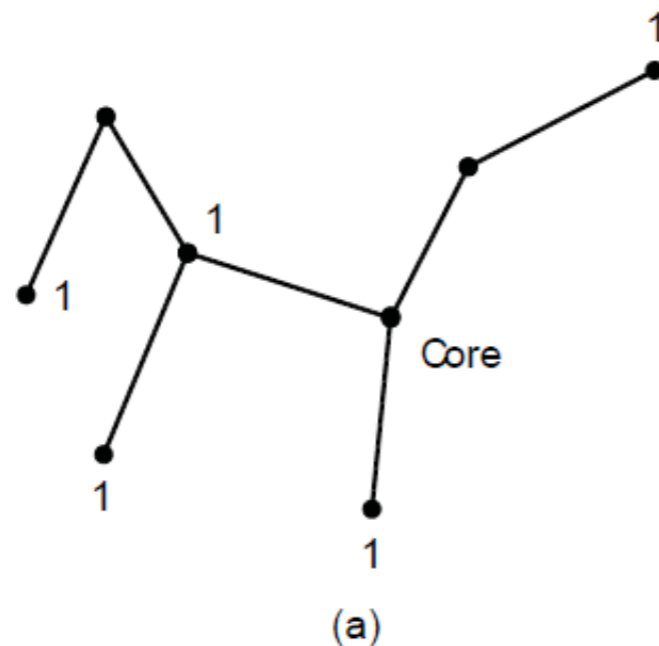


(c)

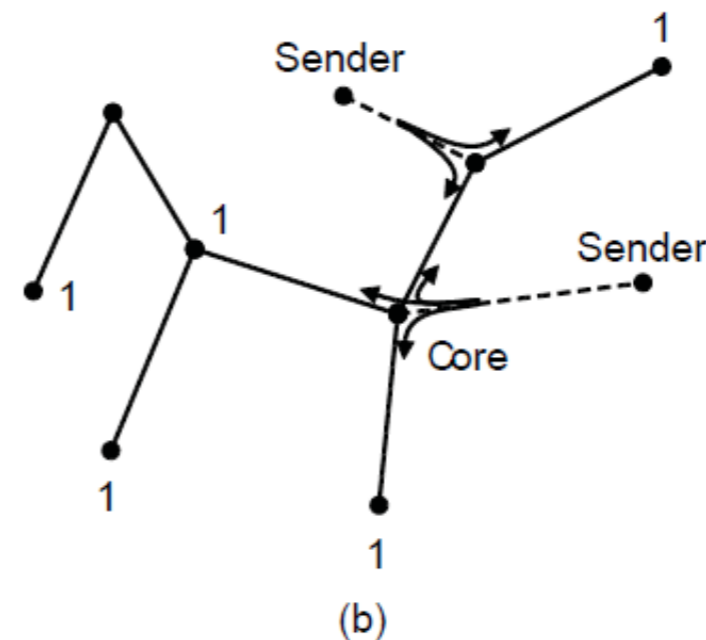
(d)

# Multicast Routing

- Core-based trees
  - Instead of all routers calculating their own multicast tree
  - Create a single one from one randomly selected core
  - Sender sends to core, core distributes
  - Short-cut if message reaches the multicast tree



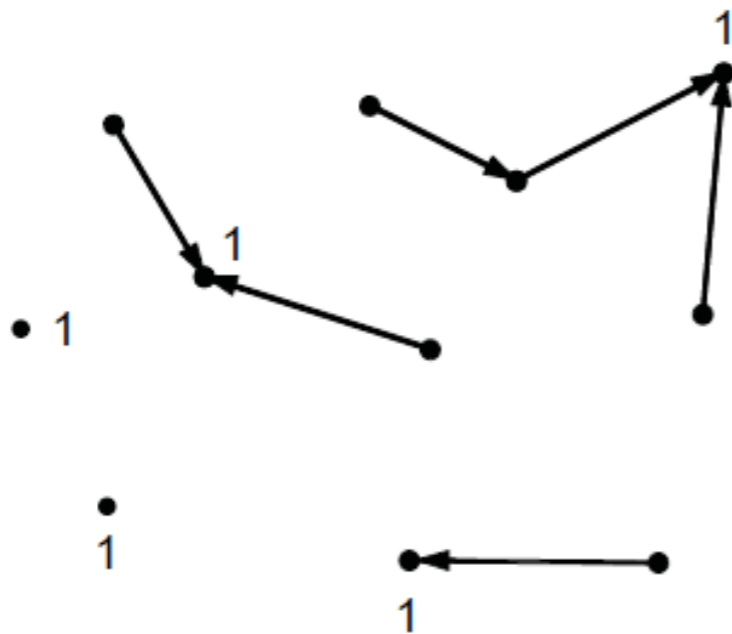
Sink tree from core



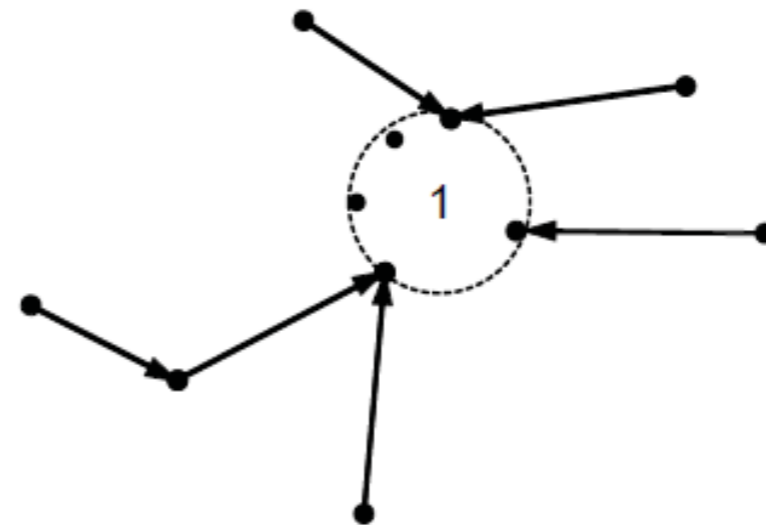
Multicast is sent to the core until it reaches the sink tree

# Anycast Routing

- Anycast:
  - Goes to the first node in a group
  - Dijkstra's algorithm will generate good sink trees



Anycast routes to group 1

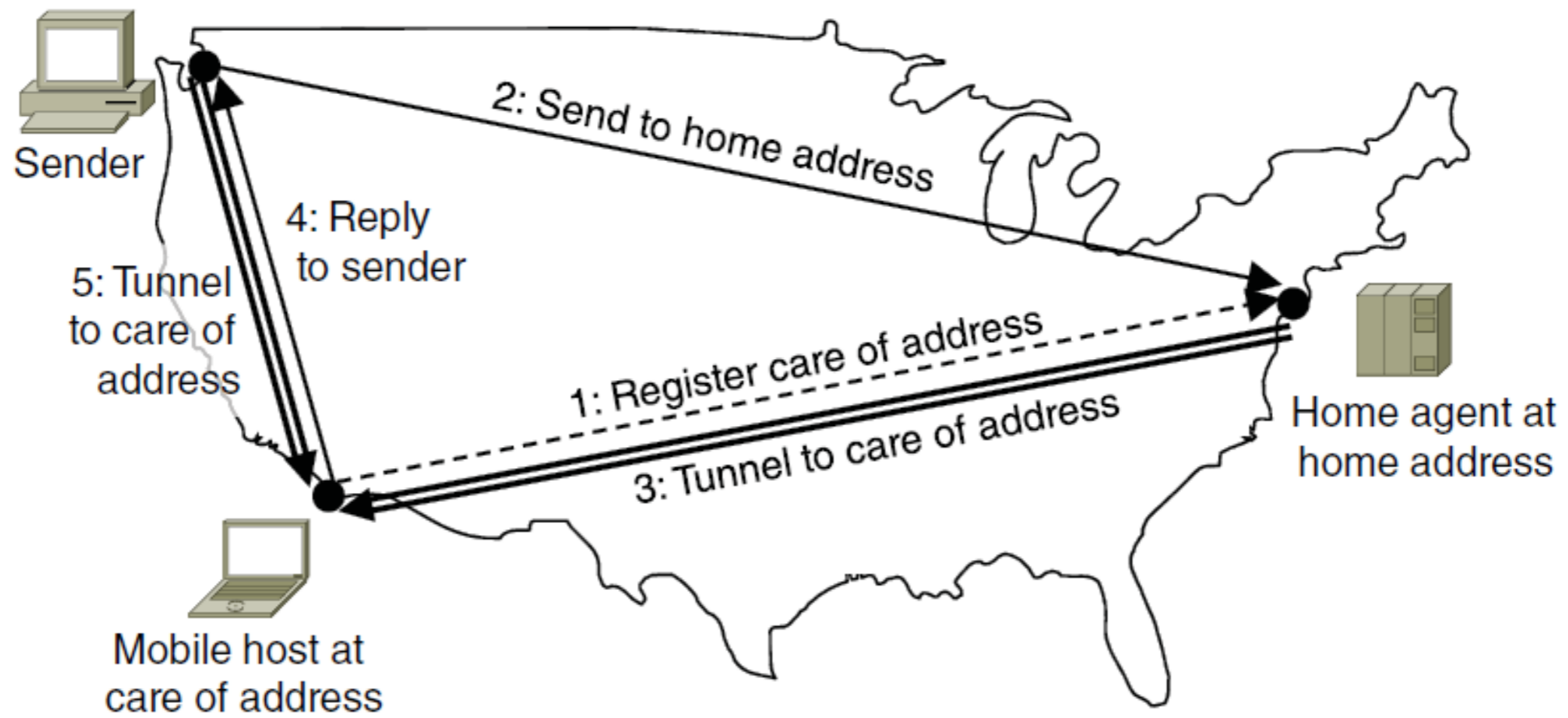


Apparent topology of sink tree to "node" 1

# Anycast Routing

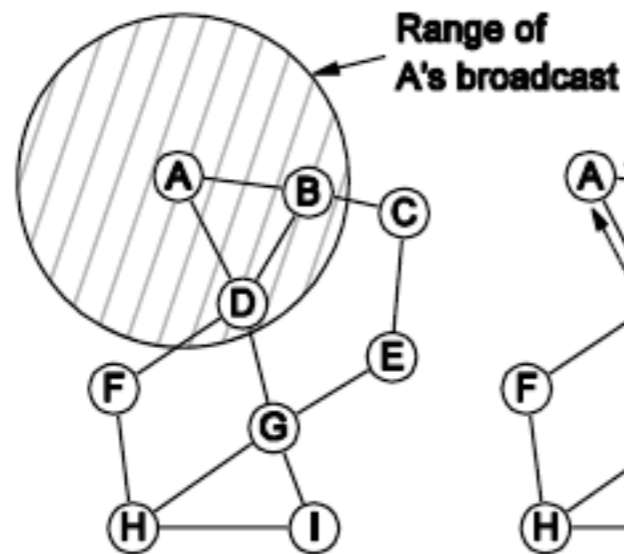
- Anycast Example:
  - A collection of servers has the same IPv6 address
    - Routers advertise routes to the IPv6 address
    - A router will send the request to the nearest server with the address
    - Local router(s) know that this is an anycast address
  - Potential rerouting of a long-term TCP connection has not been observed in practice.

# Routing for Mobile Hosts: Home Agents

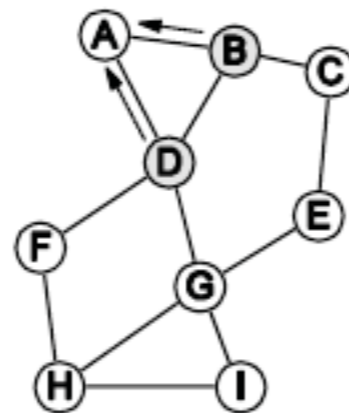


# Routing in Ad-Hoc Networks

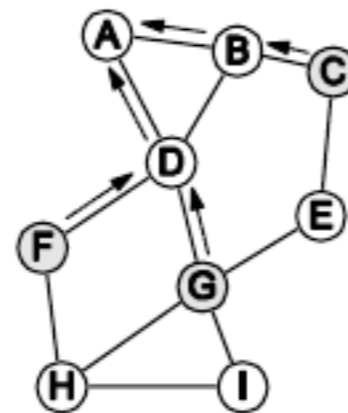
- Network technology changes as wireless nodes move
- Routes are made on demand
  - Ad hoc On-demand Distance Vector (AODV)



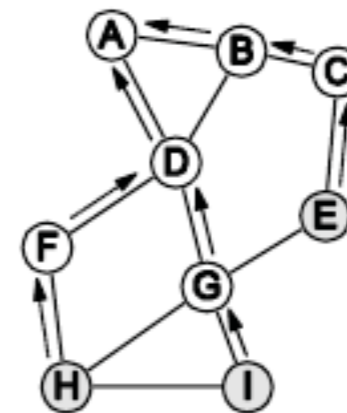
*A's starts to find route to I*



*A's broadcast reaches B & D*



*B's and D's broadcast reach C, F & G*



*C's, F's and G's broadcast reach H & I*

# Congestion Control

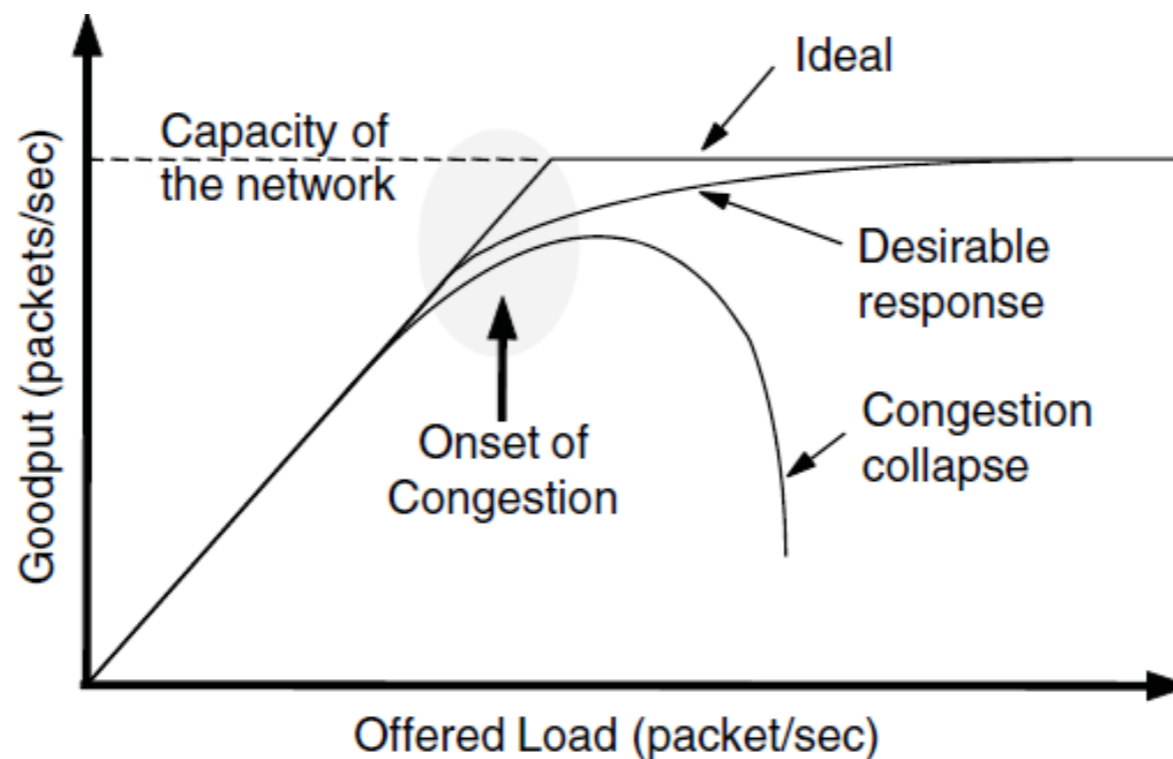


# Congestion Control

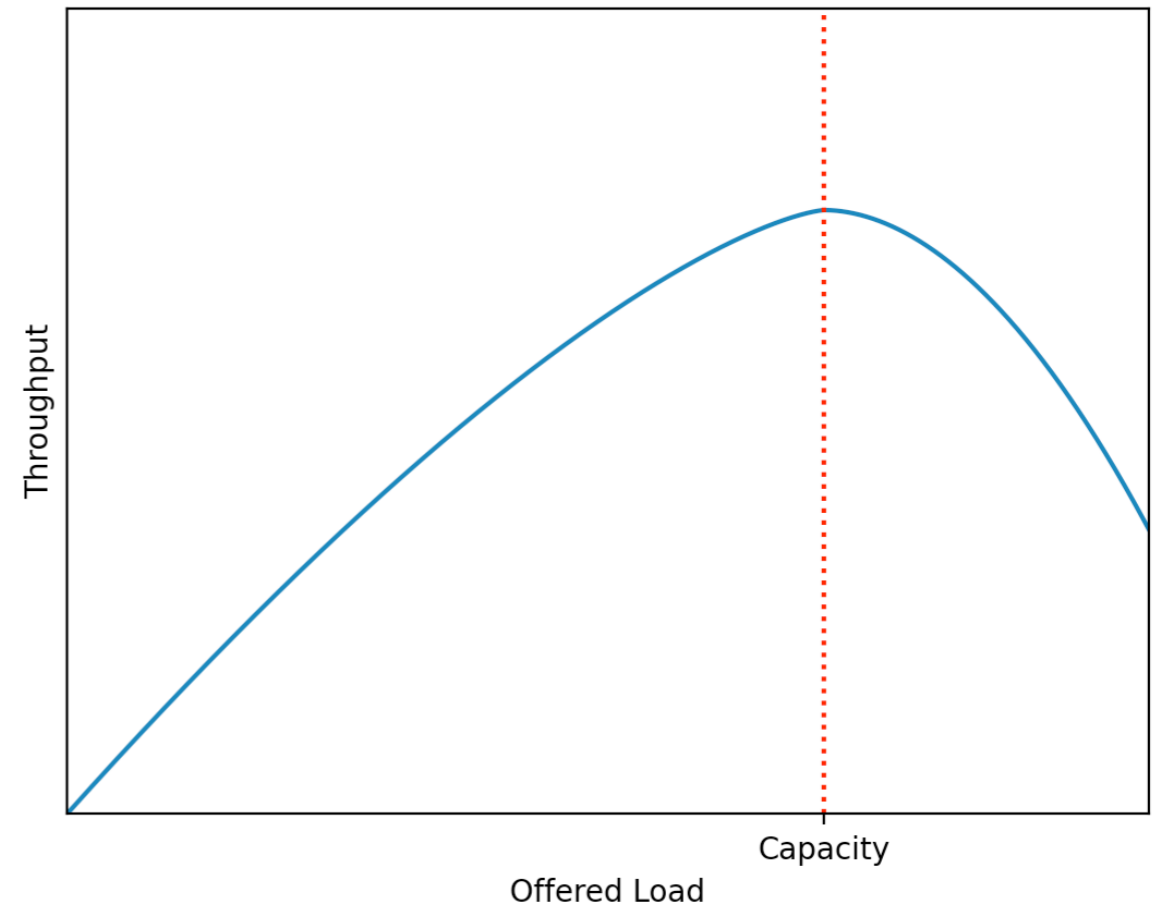
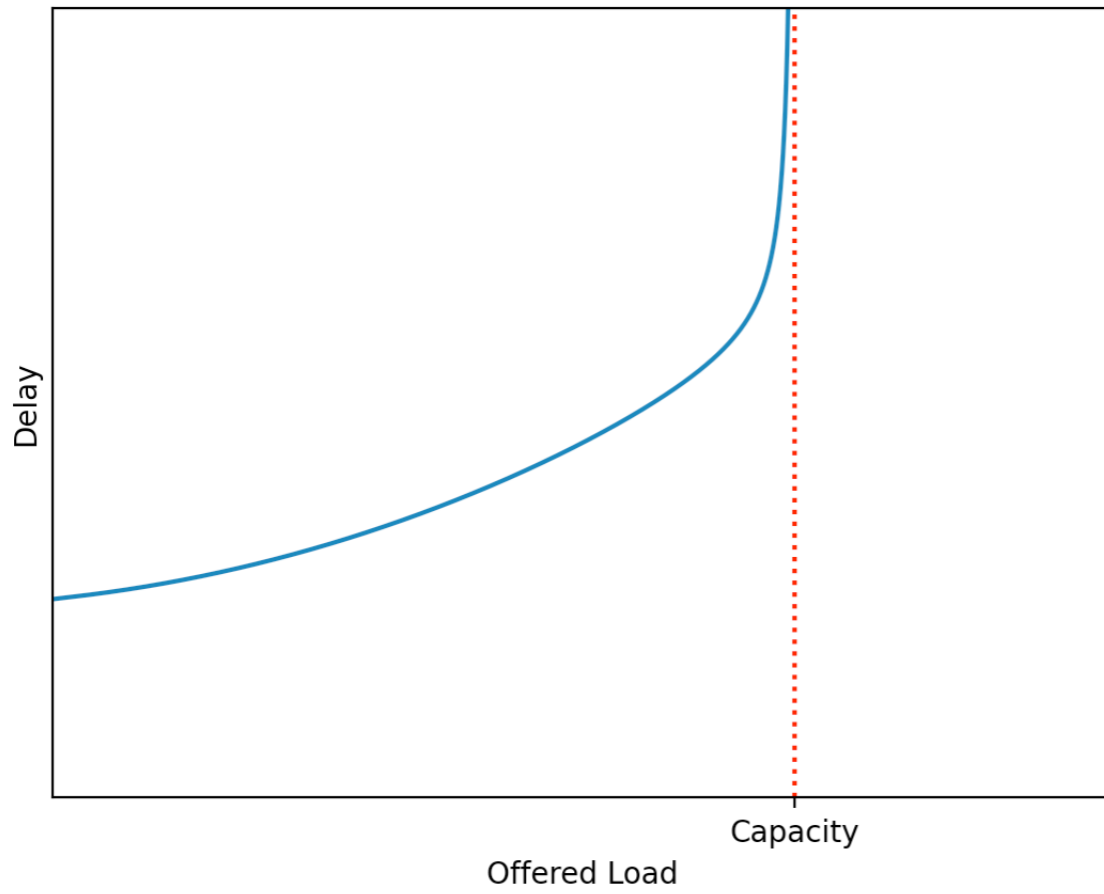
- Handling congestion is the responsibility of the Network and Transport layers working together
  - We look at the Network portion here

# Congestion Control

- Congestion results when too much traffic is offered
  - Performance degrades
    - due to loss / retransmissions
  - Goodput ( useful packets) trails offered load

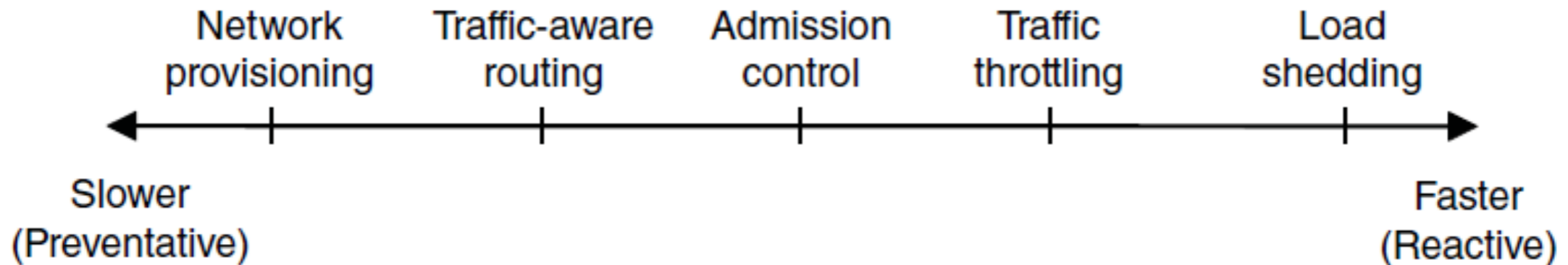


# Congestion Control



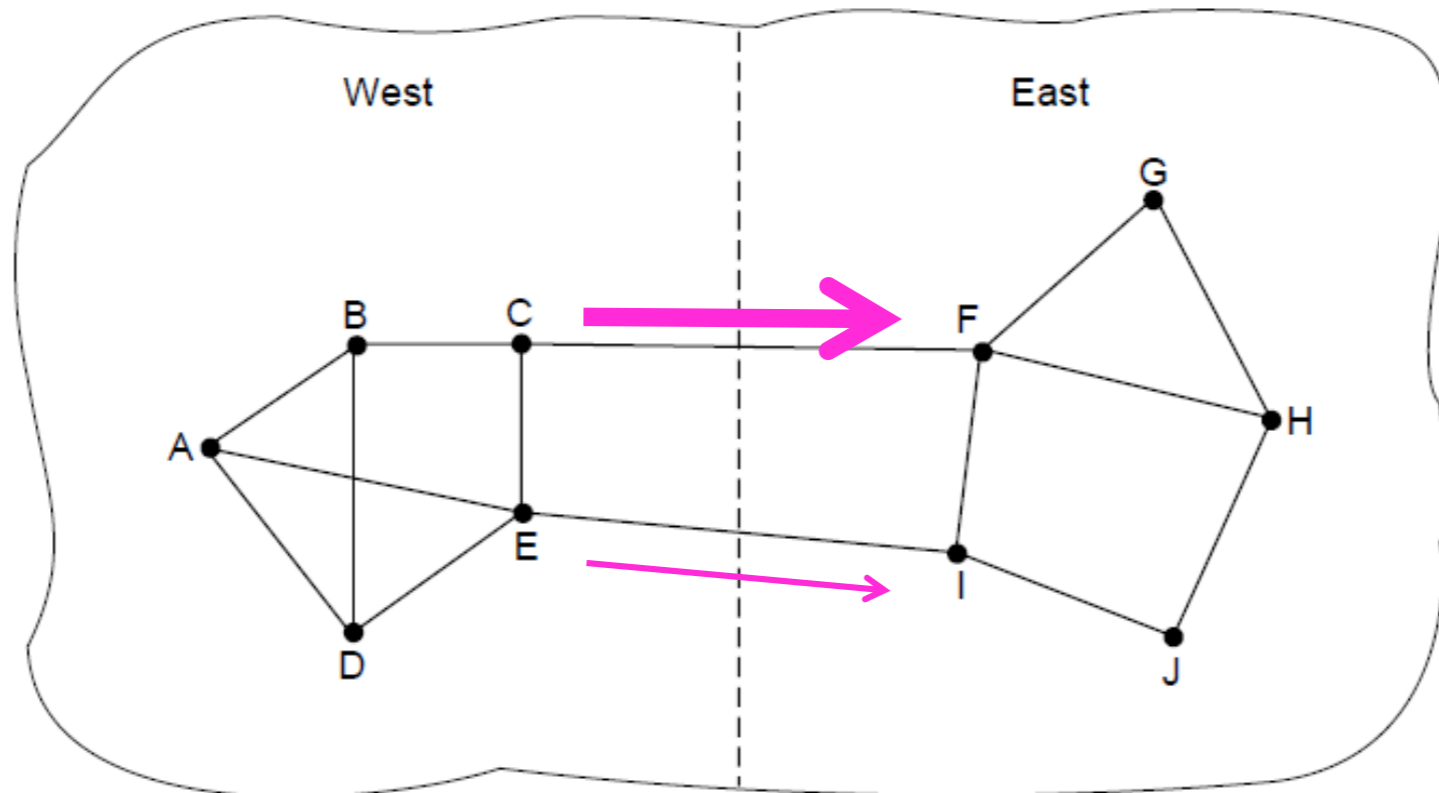
# Congestion Control

- Network must do its best with the offered load
  - Different approaches at different timescales
  - Nodes need to reduce offered load (transport layer)



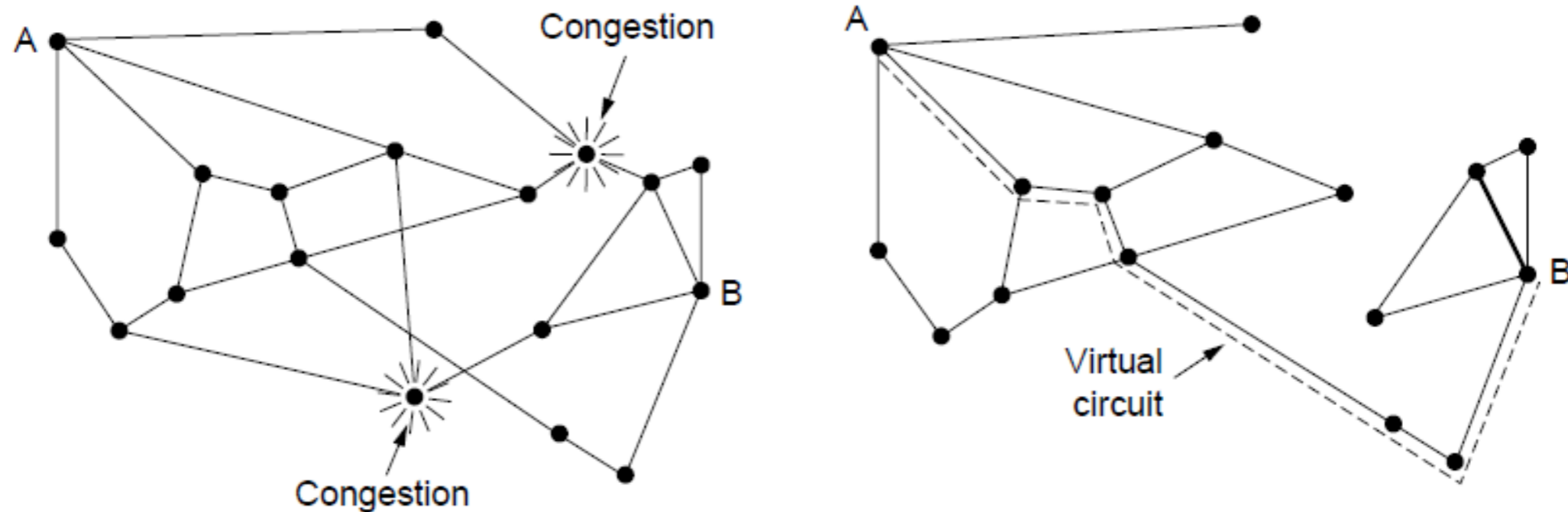
# Traffic Aware Routing

- Choose routes depending on traffic, not just topology
  - Below, use EI for West-to-East traffic if link CF is loaded
  - Need to avoid oscillations
    - Change routes slowly by adjusting weights
    - Use multiple paths in routing tables



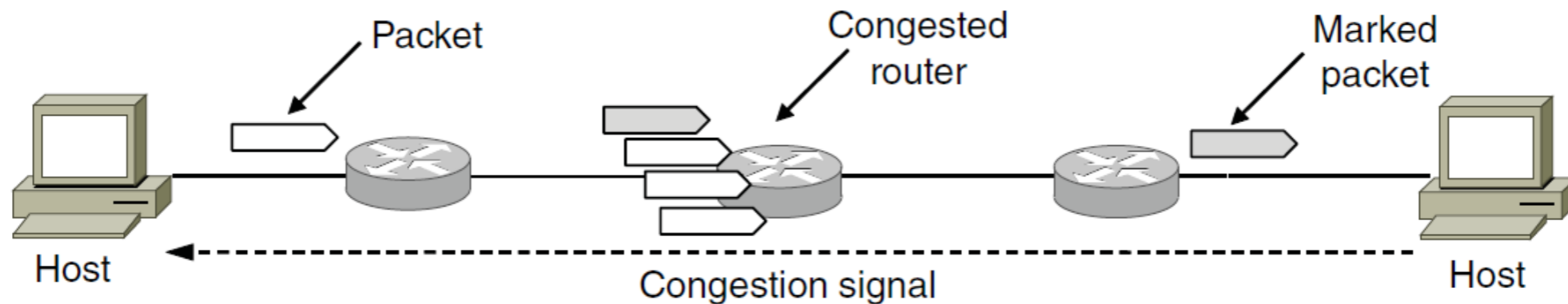
# Admission Control

- Allow new traffic load only if the network has sufficient capacity
  - Easy to do with virtual circuits
    - Can look for an uncongested route in the set-up phase



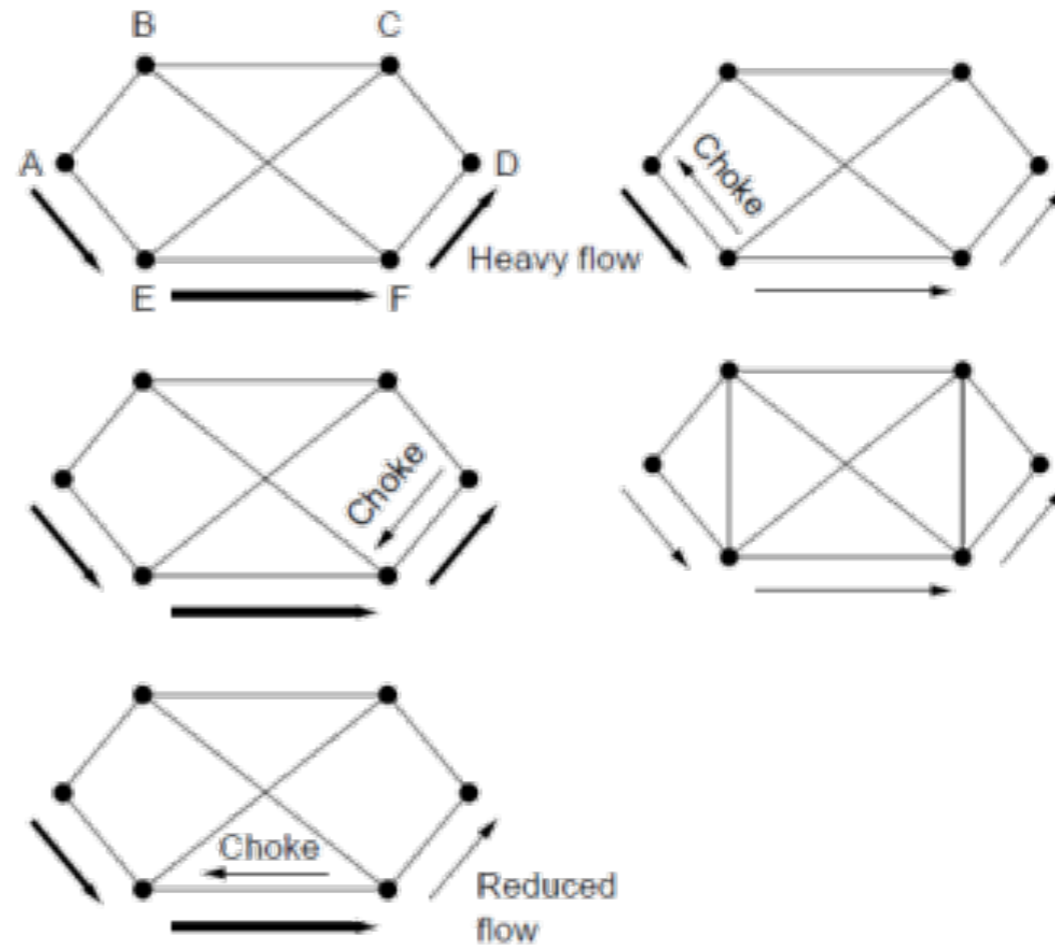
# Traffic Throttling

- Congested routers signal hosts to slow down traffic
  - ECN (Explicit Congestion Notification) marks packets and receiver returns signal to sender



# Load Shedding

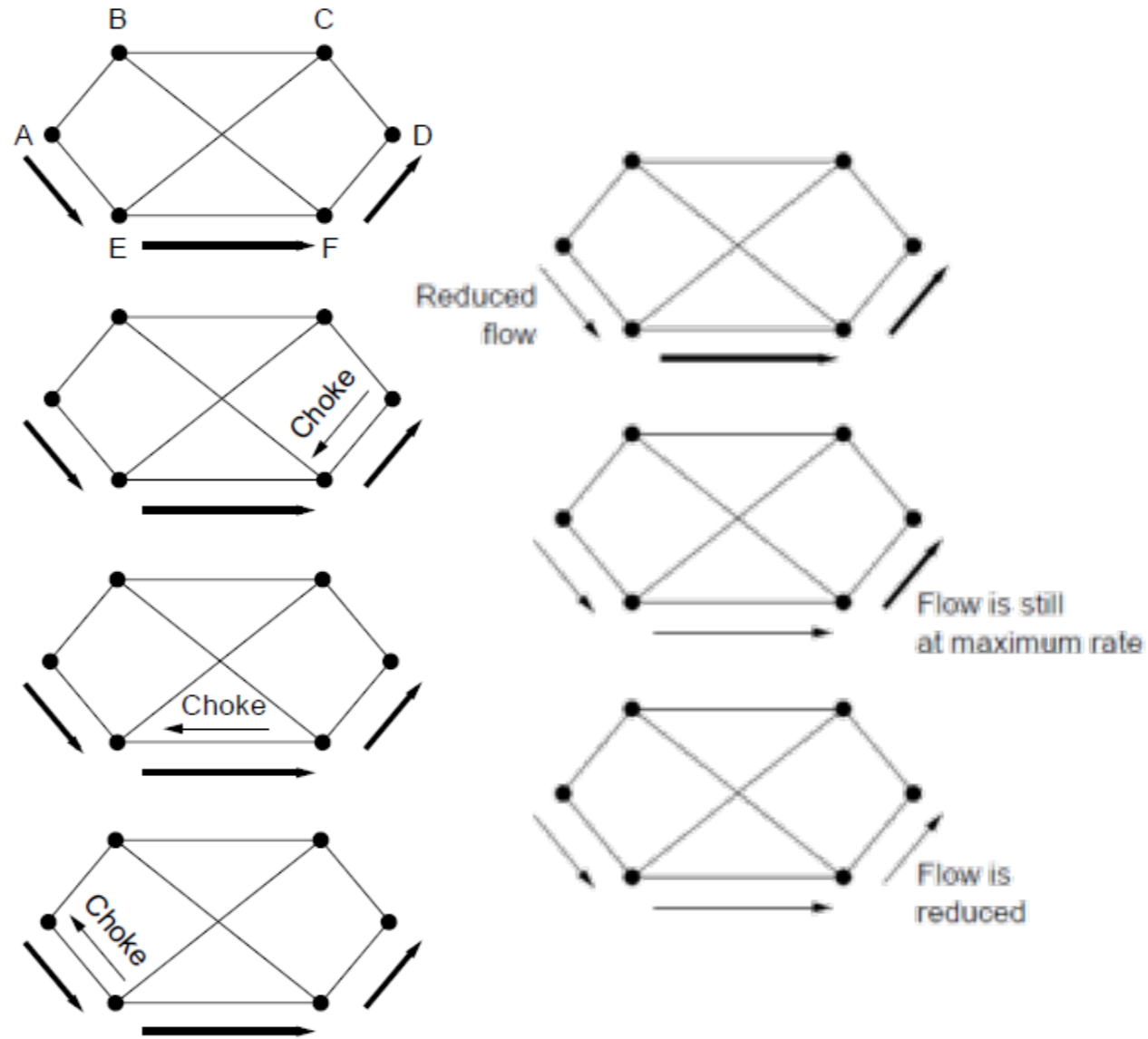
- When all else fails, network will drop packets (shed load)
- Can be done end-to-end or link-by-link
- Link-by-link (right) produces rapid relief





# Load Shedding

- End-to-end (right) takes longer to have an effect, but can better target the cause of congestion





# Quality of Service

# Quality of Service

- Different applications care about different properties
  - We want provide all applications which what they need

<b>Application</b>	<b>Bandwidth</b>	<b>Delay</b>	<b>Jitter</b>	<b>Loss</b>
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

# Quality of Service

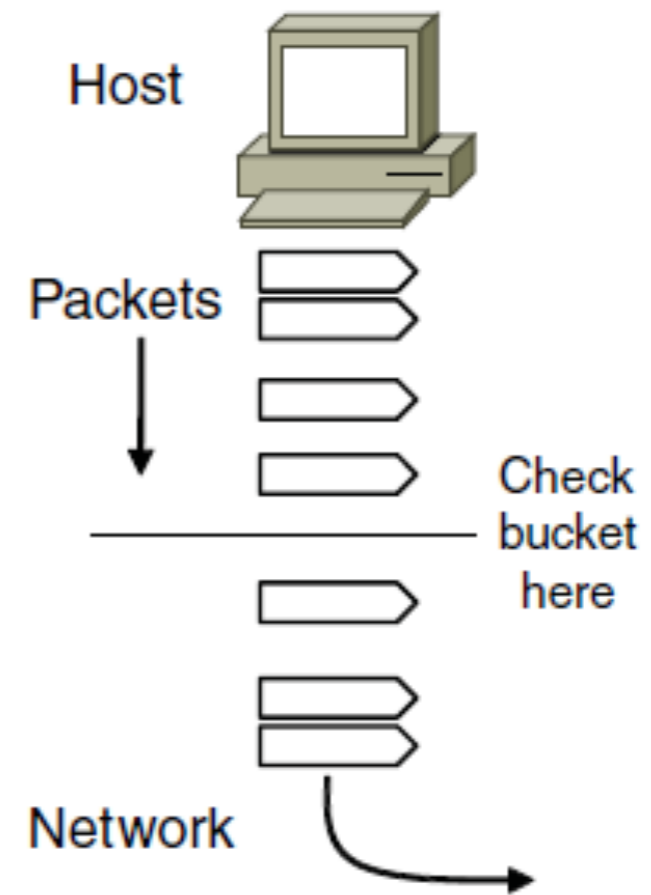
- Network provides service with different kinds of QoS (Quality of Service) to meet application requirements

<b>Network Service</b>	<b>Application</b>
Constant bit rate	Telephony
Real-time variable bit rate	Videoconferencing
Non-real-time variable bit rate	Streaming a movie
Available bit rate	File transfer

Example of QoS categories from ATM networks

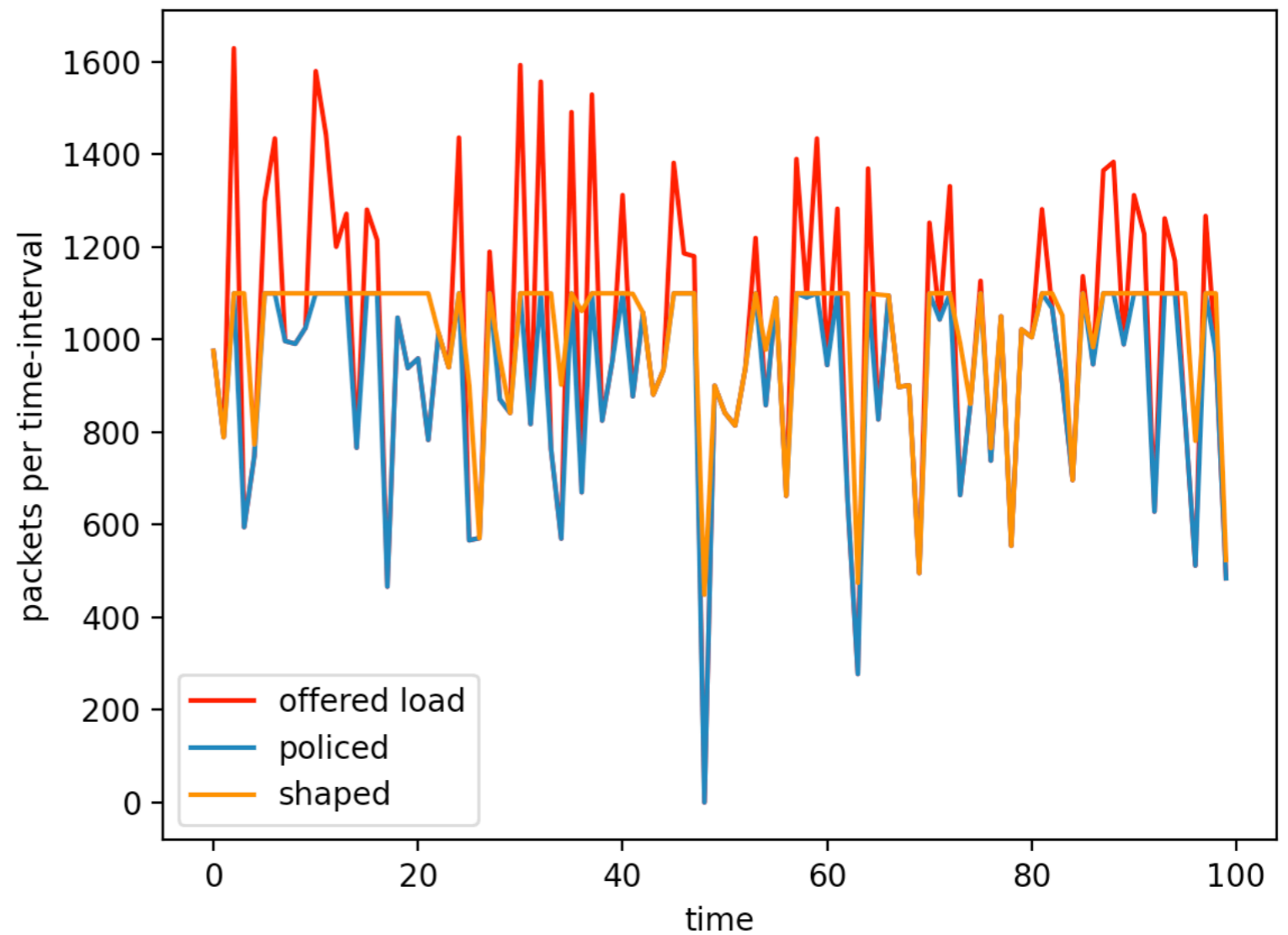
# Traffic Shaping

- Traffic shaping regulates the average rate and burstiness of data entering the network
- Allows guarantees



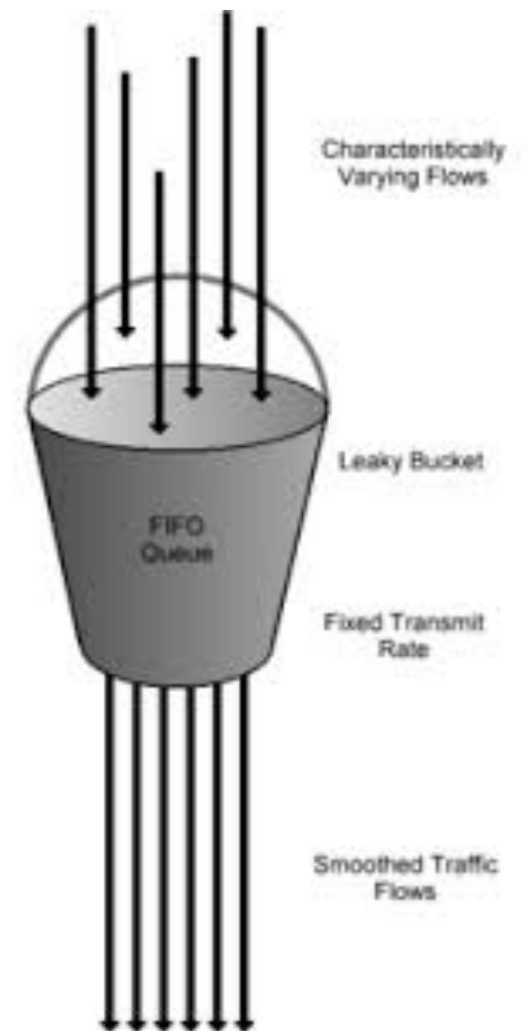
# Traffic Shaping

- Policing:
  - Drop packets if there are too many
- Shaping:
  - delay packets (up to a point)



# Traffic Shaping

- Leaky bucket
  - Limits average rate and short-term burst of traffic
    - Water enters at variable rate
    - If there is water, leaves at fixed rate
    - If the bucket overflows, you are dropping



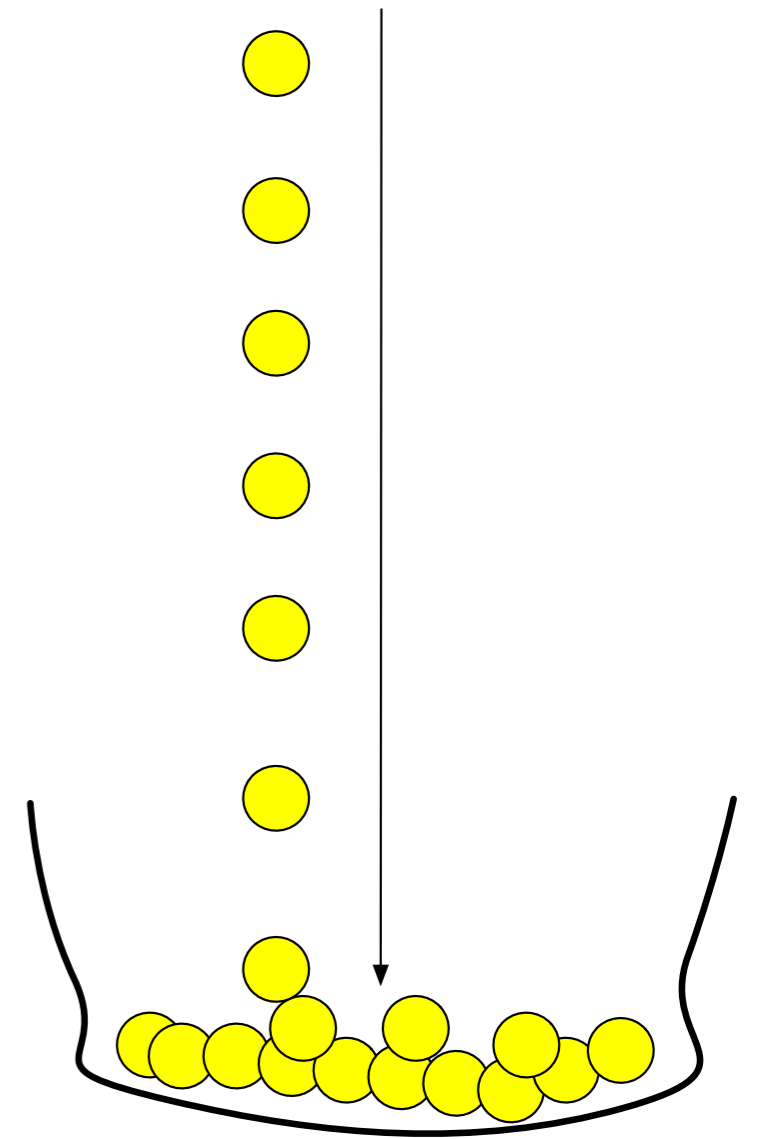
# Traffic Shaping

- Token bucket
  - Allows bursts while still preventing monopolizing a link

Generate tokens at fixed rate

Place token in bucket

Take a token from bucket for each packet send





# Traffic Shaping

- Both leaky bucket and token bucket allow bursts but limit long-term rates

# Traffic Shaping

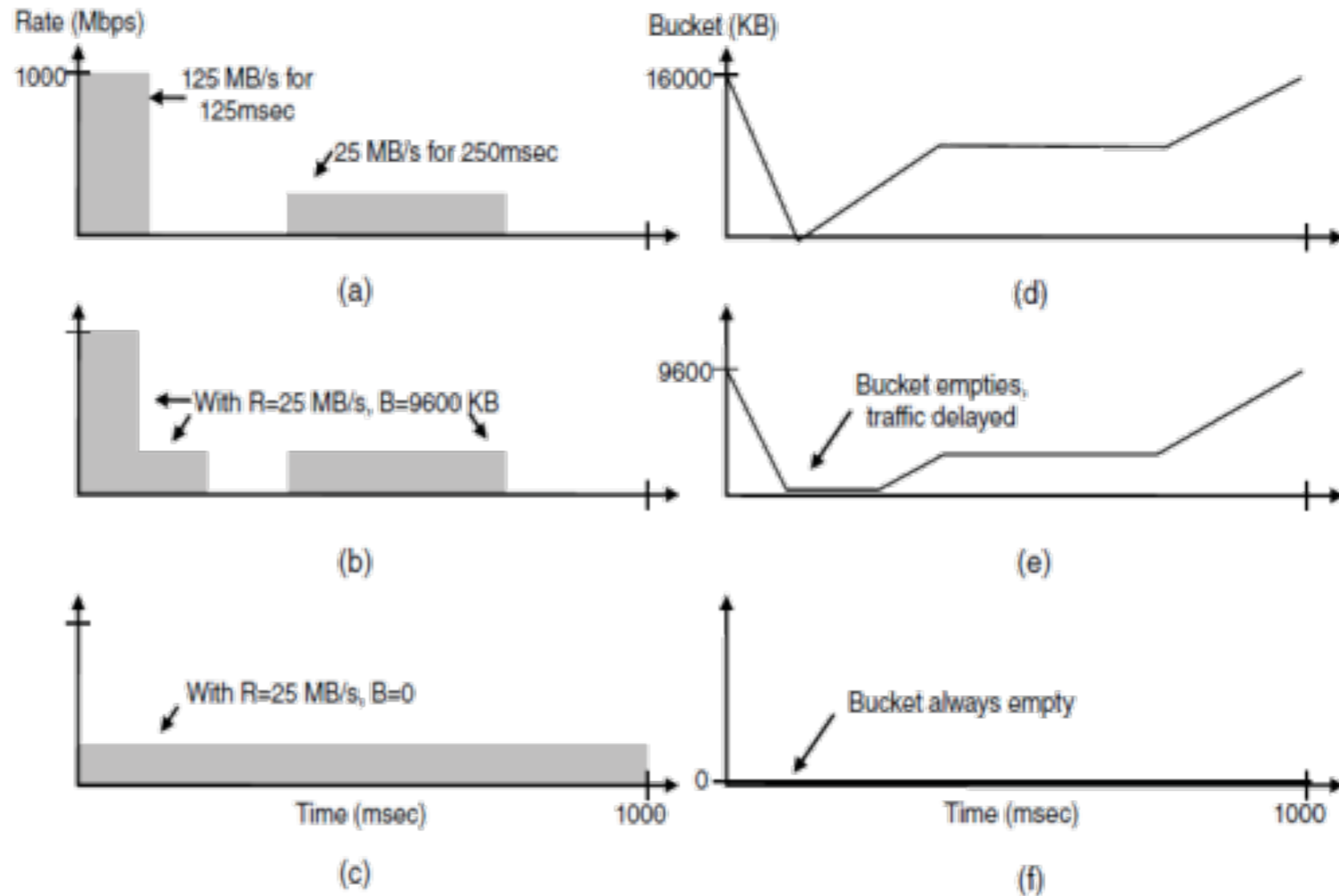
Host traffic  
 $R=200$  Mbps  
 $B=16000$  KB



Shaped by  
 $R=200$  Mbps  
 $B=9600$  KB



Shaped by  
 $R=200$  Mbps  
 $B=0$  KB

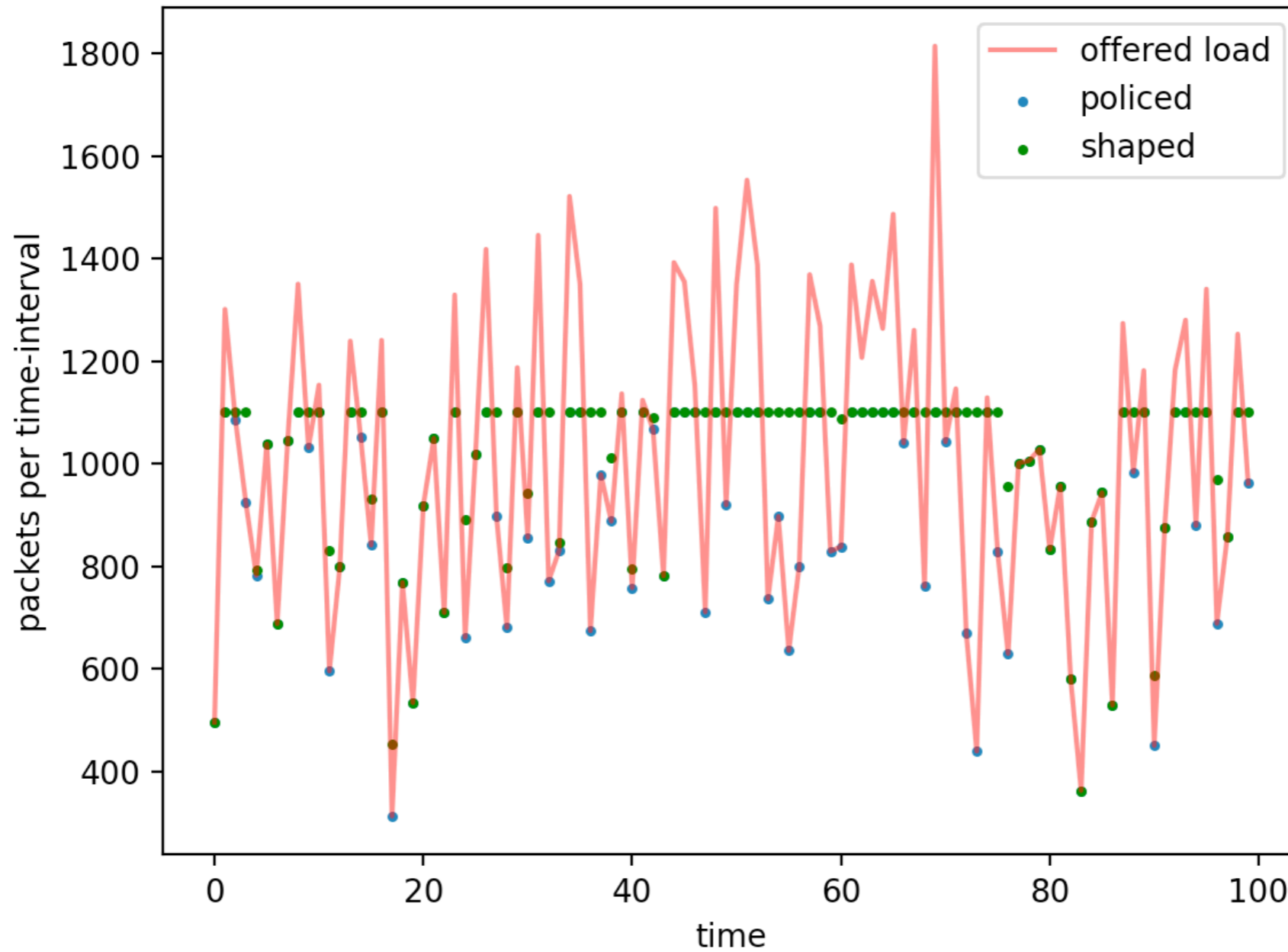


Token Bucket: Smaller bucket size delays traffic, but reduces burstiness

# Traffic Shaping

- Examples:
  - 1100 Tokens per time-unit
  - Time 0: Load: 975 — send all
  - Time 1: Load 790 — send all
  - Time 2: Load 1630 — send 1100, keep 530 buffered
  - Time 3: Load 595 — send 540 buffered plus 560 new ones
    - 25 are still buffered
  - Time 4: Load 1435: send 25 buffered plus 1075 new ones
    - 360 are buffered
  - Time 5: Load 997: send 360 buffered plus 740 new ones
    - 257 are buffered

# Traffic Shaping

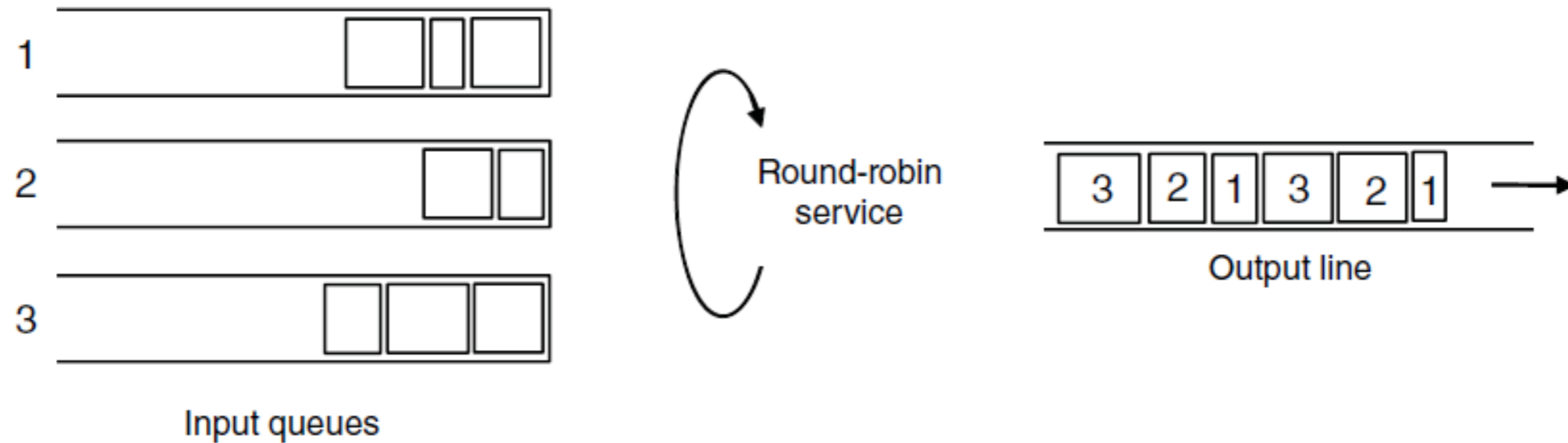


Load is  $\sim N(1000, 300)$ , token rate is 1100

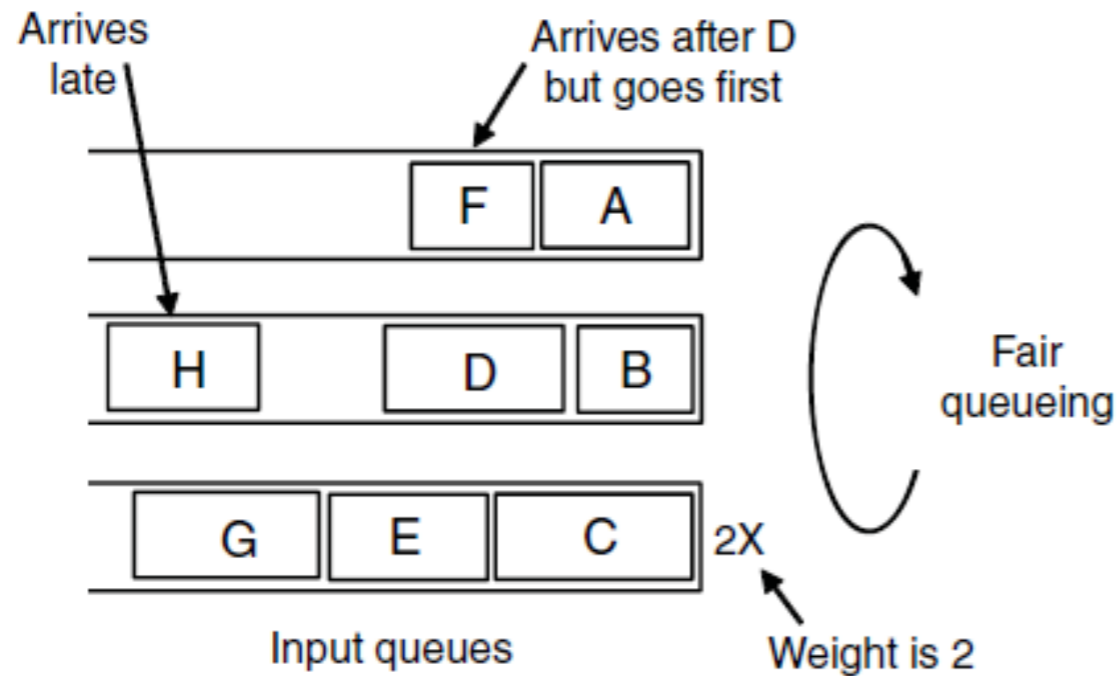
# Packet Scheduling

- Assumes fixed routes
  - Reserves resources for different flows
    - Bandwidth, Buffer space, CPU cycles
- Can impose different queuing disciplines
  - FIFO : First In, First Out = FCFS : First Come First Served
  - Fair queueing
  - Weighted fair queuing

# Packet Scheduling: Round Robin Queueing



# Weighted Fair Queuing



Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

Send in rounds

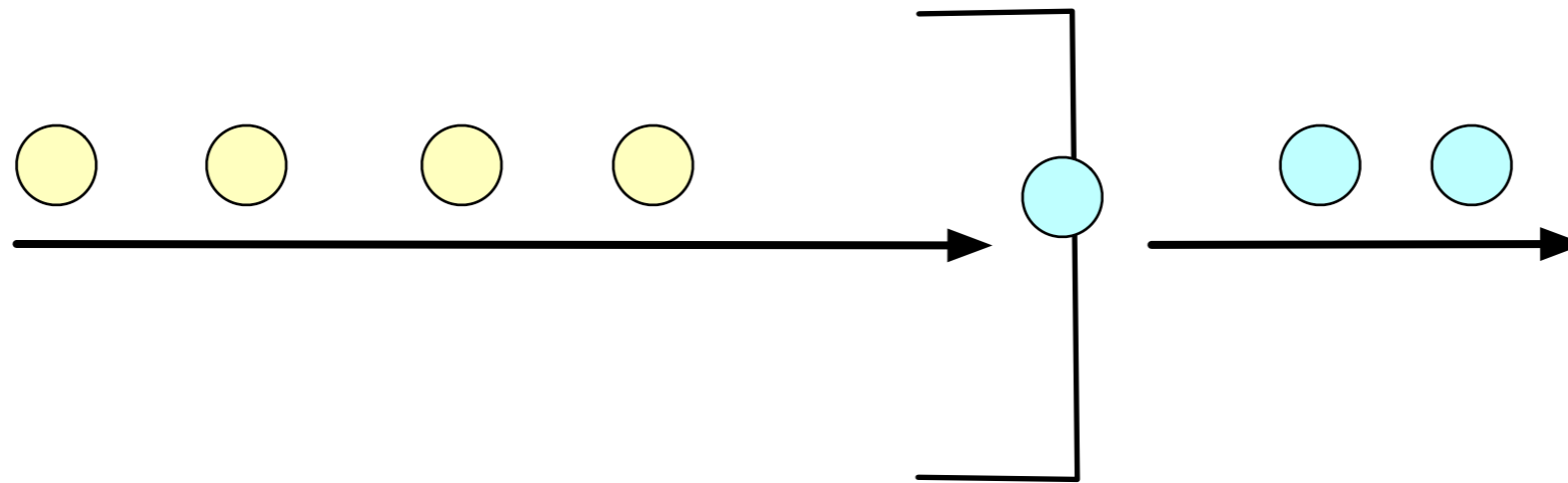
Weight tells how many from each flow get resent

# Admission Control

<b>Parameter</b>	<b>Unit</b>
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes



# Queuing Networking



- Model:
  - Clients arrive with an arrival rate of  $\lambda$ 
    - Medium time between arrivals is  $1/\lambda$
  - Receive service for a mean time of  $S = 1/\mu$
  - Clients leave
- If all times are exponentially distributed, this is a M/M/1 queue

# Queuing Networking

- Mean time between arriving at the queue and leaving the service station (sojourn time) is then

$$\frac{1}{\mu} \times \frac{1}{1 - \lambda/\mu}$$

# Group Quiz

- Packets arrive at a router at a rate of 100000 / sec.
- Router can forward packets at a rate of 150000 / sec.
  - Hence, mean service time is 6.6667  $\mu$ sec per packet
- What does the queuing network formula give as the mean sojourn time?
- What happens if the arrival rate is 10000 /sec?
- What happens if the arrival rate is 200000 / sec?

$$\frac{1}{\mu} \times \frac{1}{1 - \lambda/\mu}$$

# Answer

- $\lambda = 10^5$ ,  $\mu = 150,000$

- Mean sojourn time is 20  $\mu\text{sec}$

```
>>> Lambda = 10**5
>>> Mu = 1.5*10**5
>>> def little(l, m):
    return (1/m)*1/(1-l/m)
```

```
>>> little(Lambda, Mu)
1.99999999999999999998e-05
```

- If  $\lambda$  is 200,000, the queue is instable
- If  $\lambda$  is 10,000, the mean service time is 7.14  $\mu\text{sec}$ , just a tad above the service time of 6.67  $\mu\text{sec}$ .

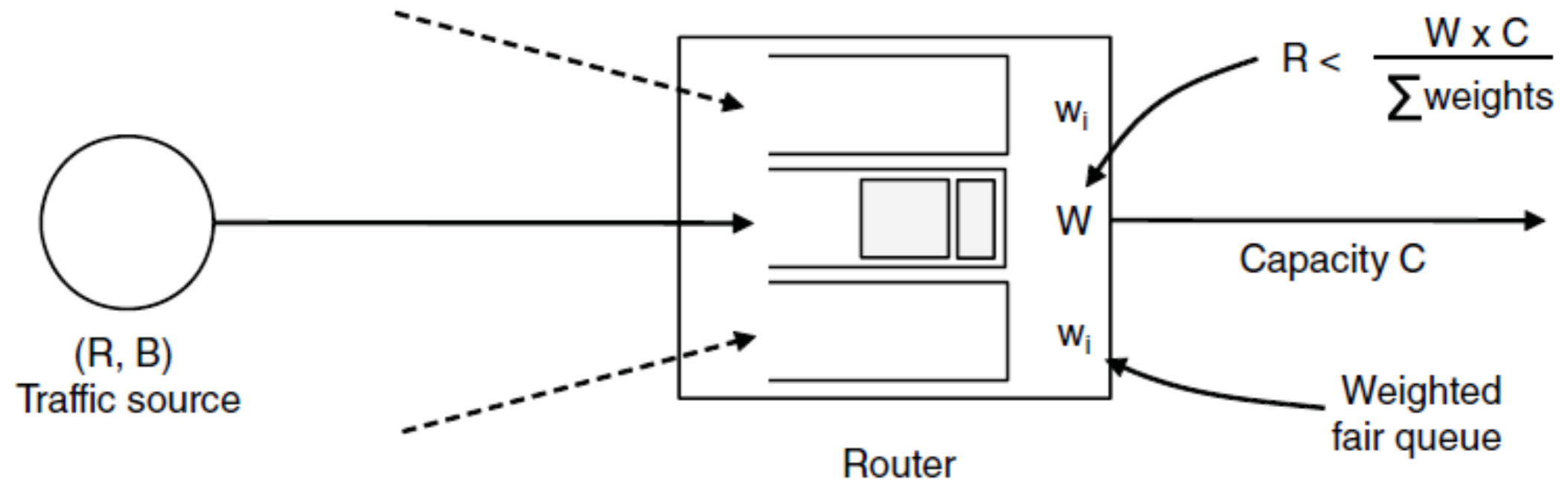
# Admission Control

- Construction to guarantee bandwidth and delay:
  - Shape traffic source to a token bucket with capacity B and bandwidth R
  - Each router runs WFQ with WFQ-weight for each flow to fulfill

$$\frac{R_i}{\text{Capacity}} < \frac{W_i}{\sum_{\text{all flows } i} W_i}$$

- This guarantees that there is enough capacity for all flows
- If this cannot be fulfilled, then do not allow the offending flow

# Admission Control



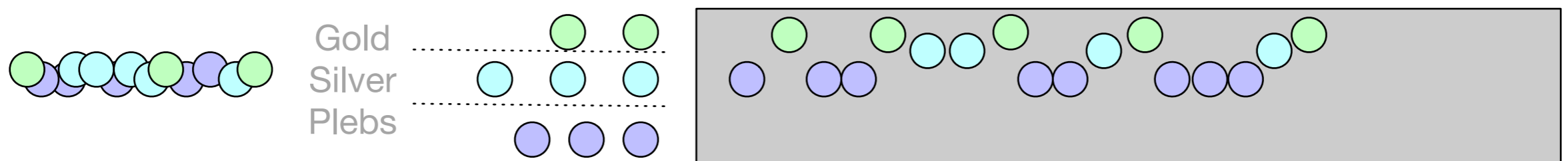
- Bandwidth is guaranteed by setting enough weight for the flow
- Delay is more subtle
  - A burst results in significant delay at the first router the flow encounters
  - But at all subsequent routers, the flow has already been shaped to be less bursty

# Individual Quiz

- Assume a router with a capacity of 200000 packets per second
- You have a flow with bandwidth requirements of 100000 packets per second. What should the weight for WFQ be?
- A second flow with bandwidth requirement of 50000 packets per second is set up. What should the weights be?
- A third flow with bandwidth requirement of 75000 packets per second is to be set up. What should the router do?

# Differentiated Services

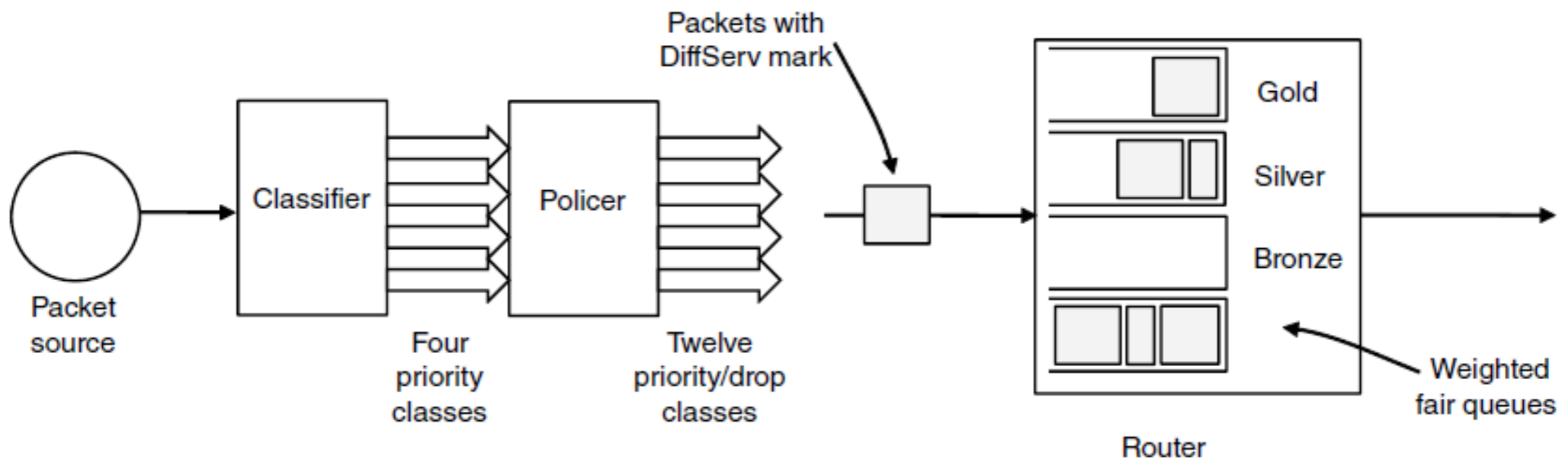
- Design classes of Quality of Service
  - Customers buy the QoS that they want
  - Packets from *expedited classes* are sent in preference to packets from regular classes





# Differentiated Service

- Implementation:
  - Customers mark desired class on packet
  - ISP shapes traffic to ensure markings are paid for
  - Router uses WFQ to provide different service levels





# Internetworking

# How networks differ

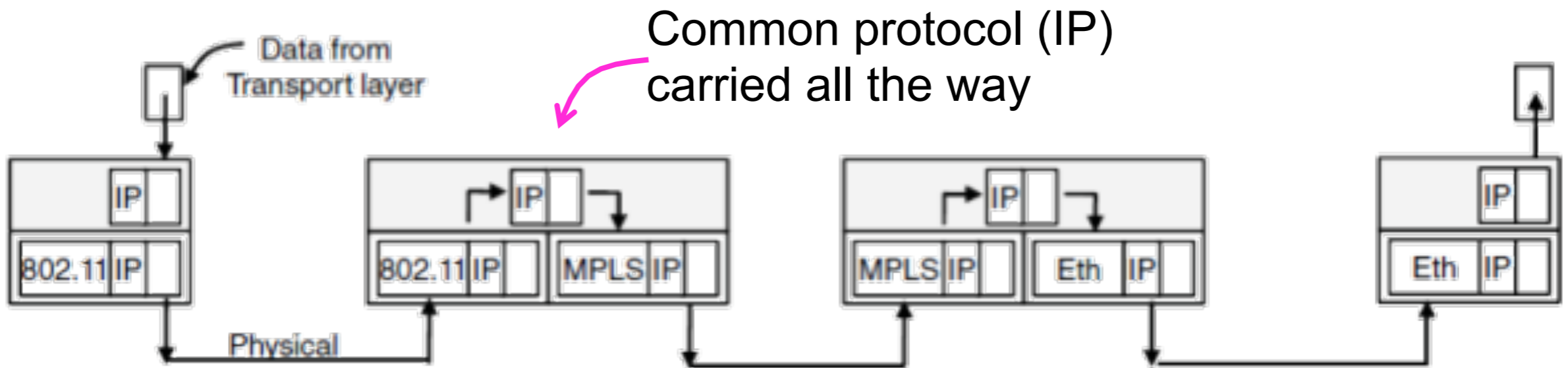
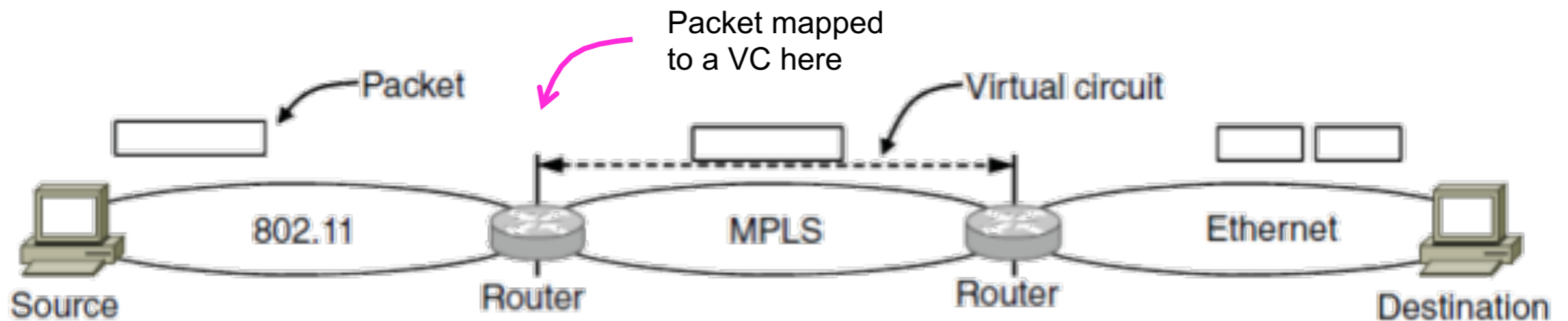
- Differences can be large:

Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

# How networks can be connected?

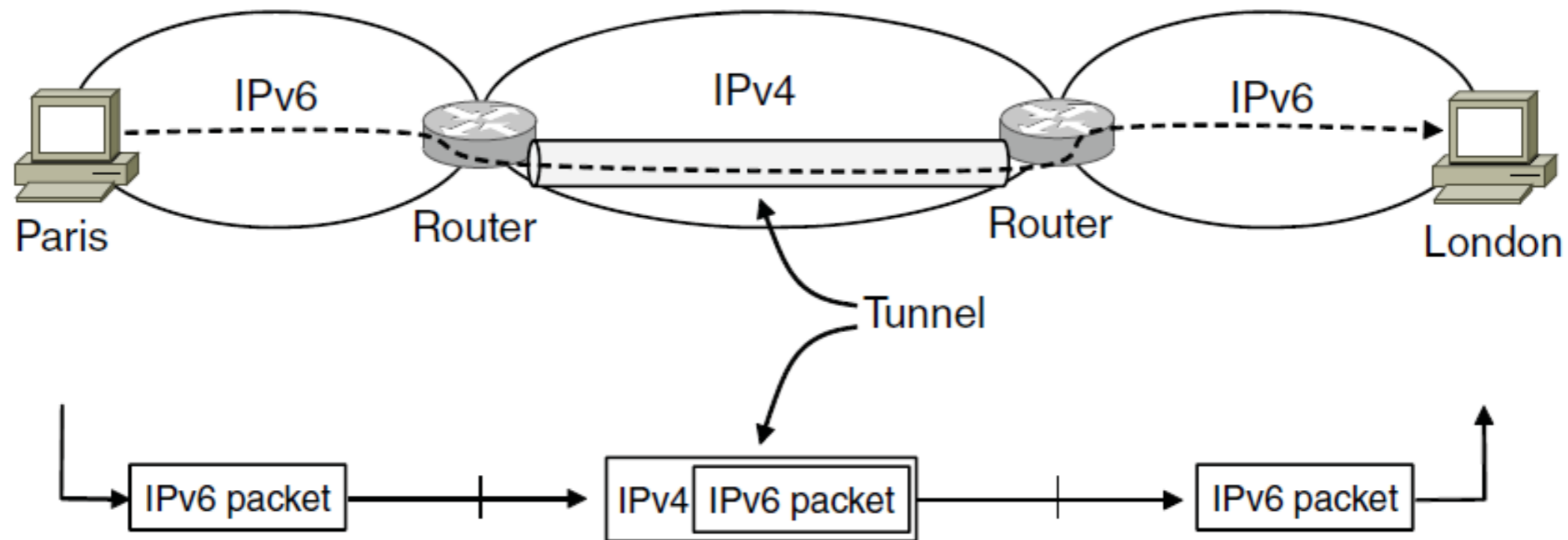
- First possibility:
  - Build translators (converters) between all types of networks
- Second possibility:
  - Build a common layer on top of the different networks
  - Cerf and Kahn (1974) —> IP

# How can networks be connected

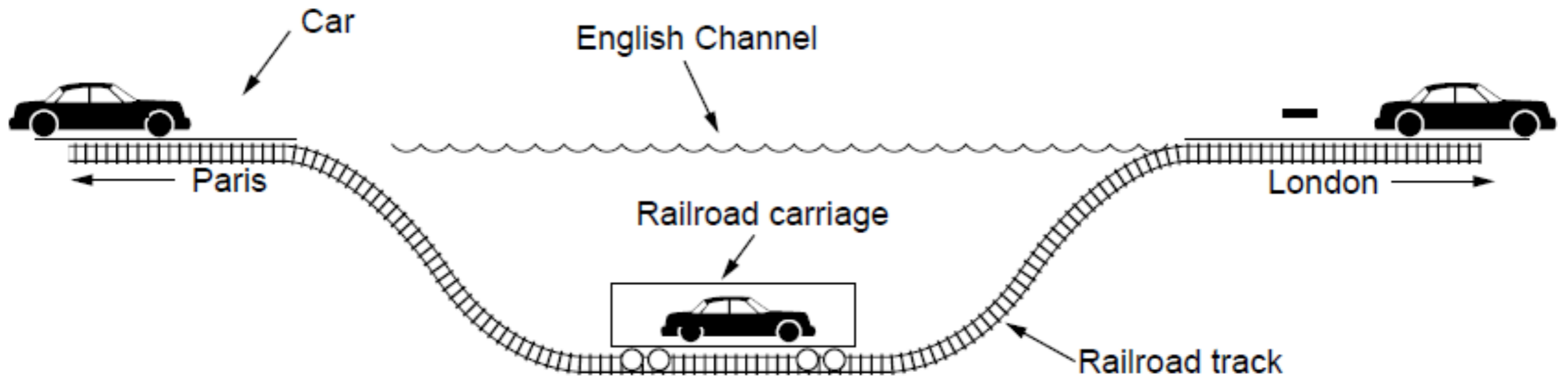


# Tunneling

- When source and destination host are on the same type of network, but there is a different network in-between:
  - Encapsulate packets over the middle



# Tunneling Analog



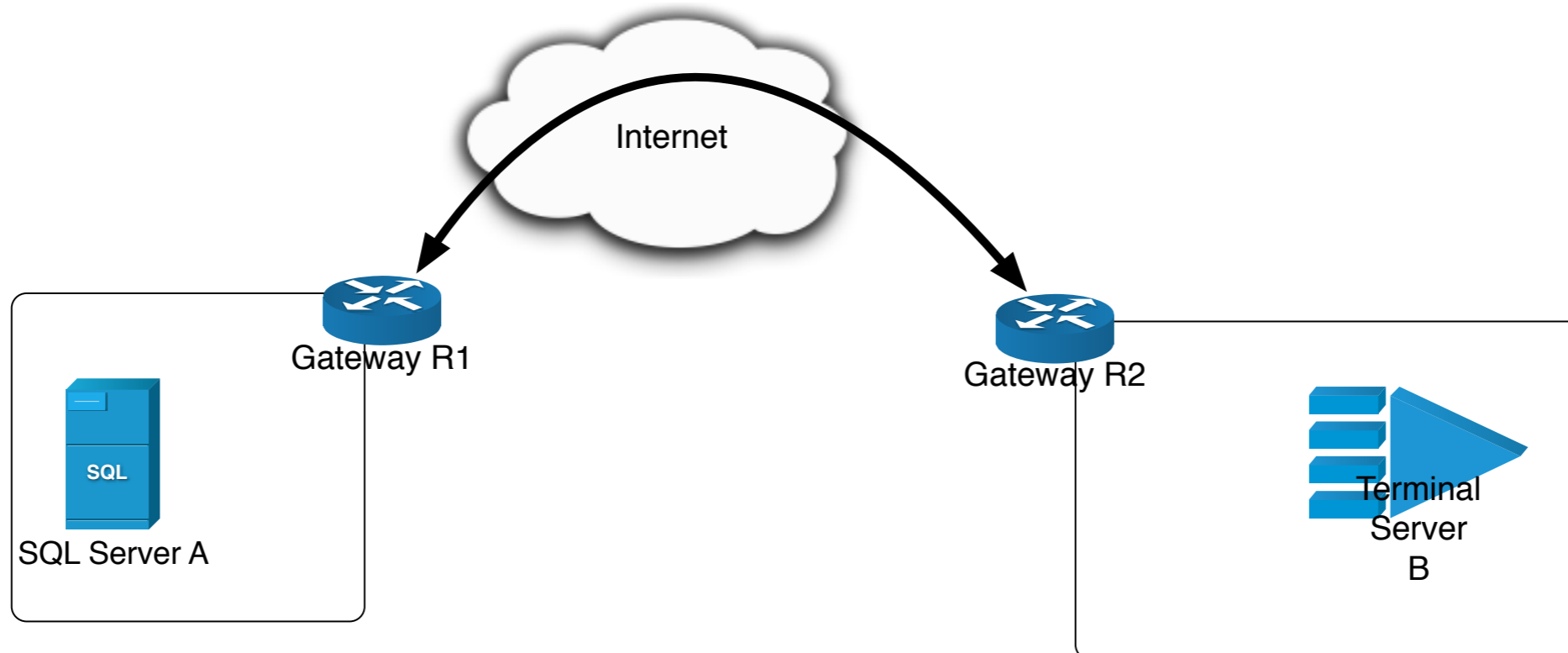


# VPN

- Virtual private networks:
  - Overlay that encrypts package contents

IPSec in Tunnel Mode

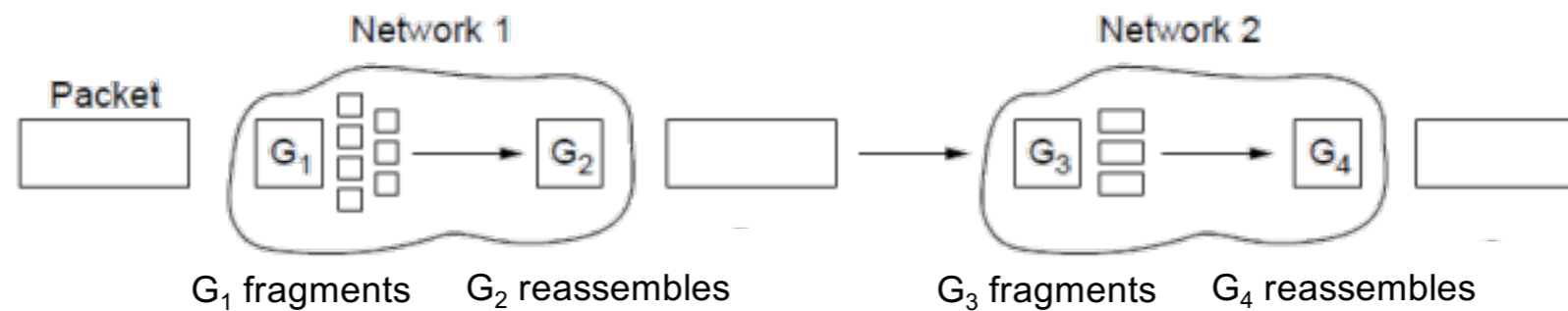
IP: src = R1, dst = R2 | ESP | {IP: src=A, dst=B | payload}



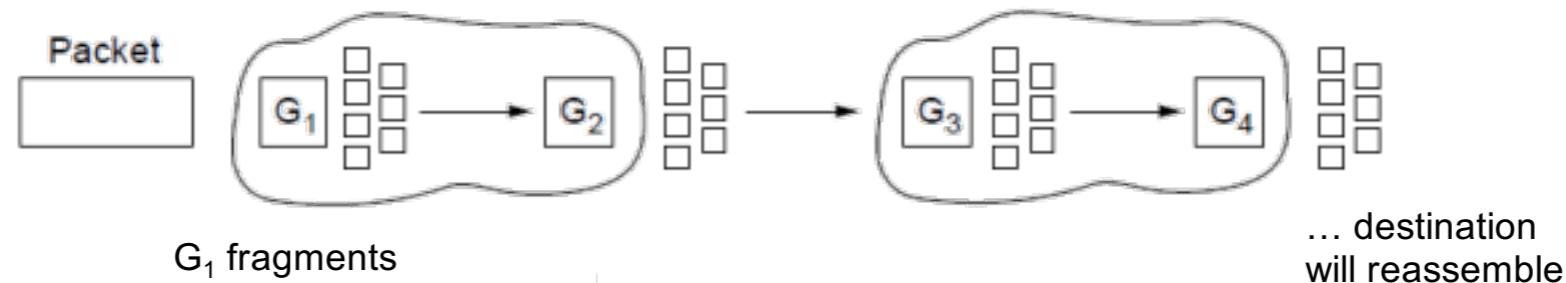


# Fragmentation

- Networks have different packet size limits
  - Packets that are too large are broken into fragments
  - IP reassembles at the destination

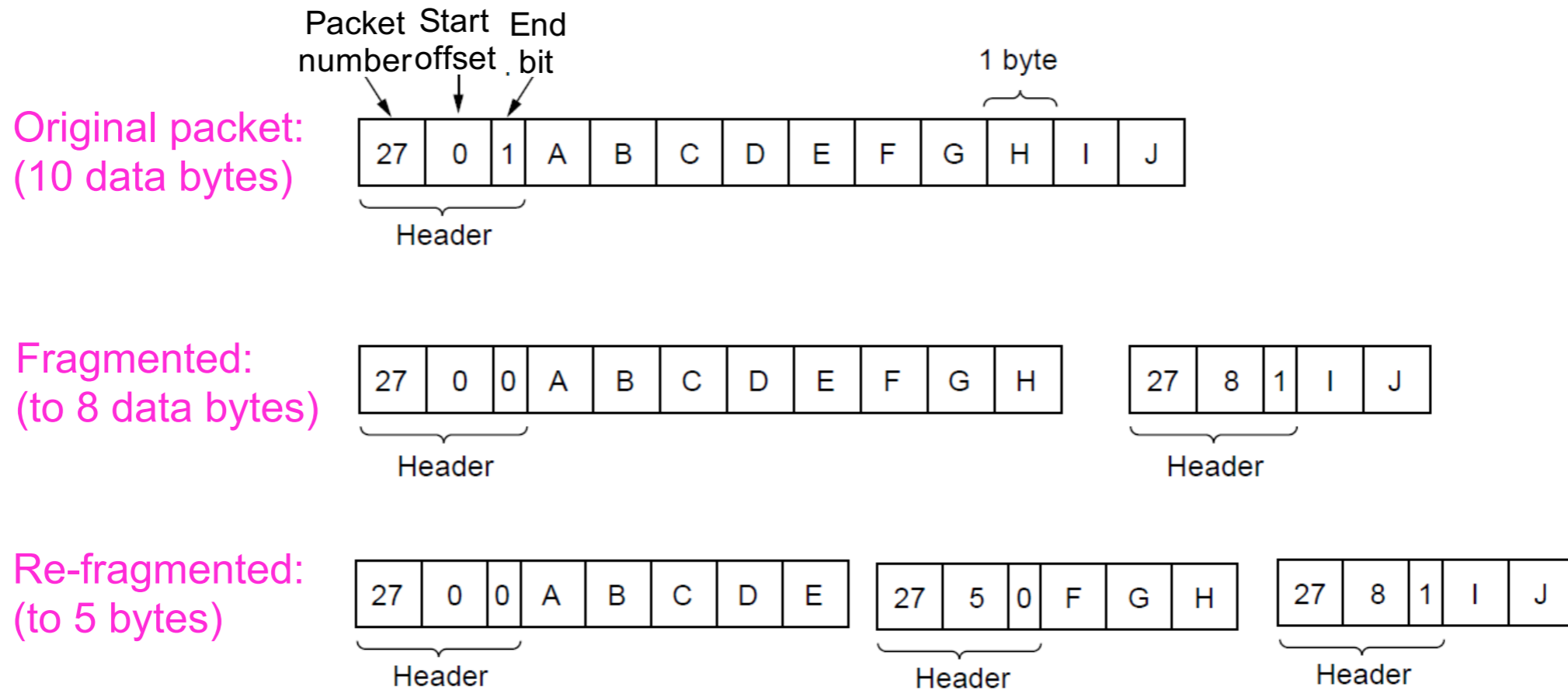


Transparent – packets fragmented / reassembled in each network



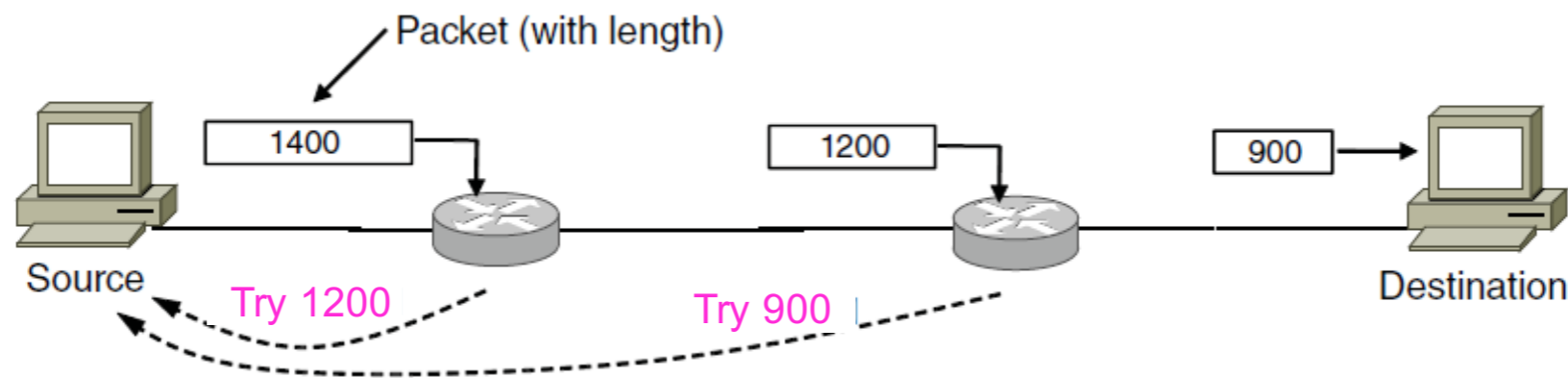
Non-transparent – fragments are reassembled at destination

# Packet fragmentation in IP



# Path MTU discovery

- Maximum Transmission Unit (MTU)
- Routers return MTU to source and discard large packets
- Can be implemented with Do Not Fragment flag and sending larger and larger packets





IP



All

News

Shopping

Books

Maps

More

Settings

Tools

About 1,310,000,000 results (0.91 seconds)

134.48.21.122

Your public IP address



Learn more about IP addresses

Feedback

### What Is My IP Address? IP Address Tools and More

[whatismyipaddress.com/](https://whatismyipaddress.com/) ▼

IP stands for Internet Protocol: The protocols are connectivity guidelines and regulations that govern computer networks. 2. IP addresses are assigned to computers, not people. The IP address you see—the one you're connected to a network and the Internet with—is assigned to the computer you're on.

[Dynamic IP vs. Static IP](#) · [Change IP](#) · [IP Tools](#) · [WAN IP Address](#) · [LAN IP ...](#)

### IP address - Wikipedia

[https://en.wikipedia.org/wiki/IP\\_address](https://en.wikipedia.org/wiki/IP_address) ▼

IP ad

An Internet F  
connected to  
communicati

People al



WHOIS

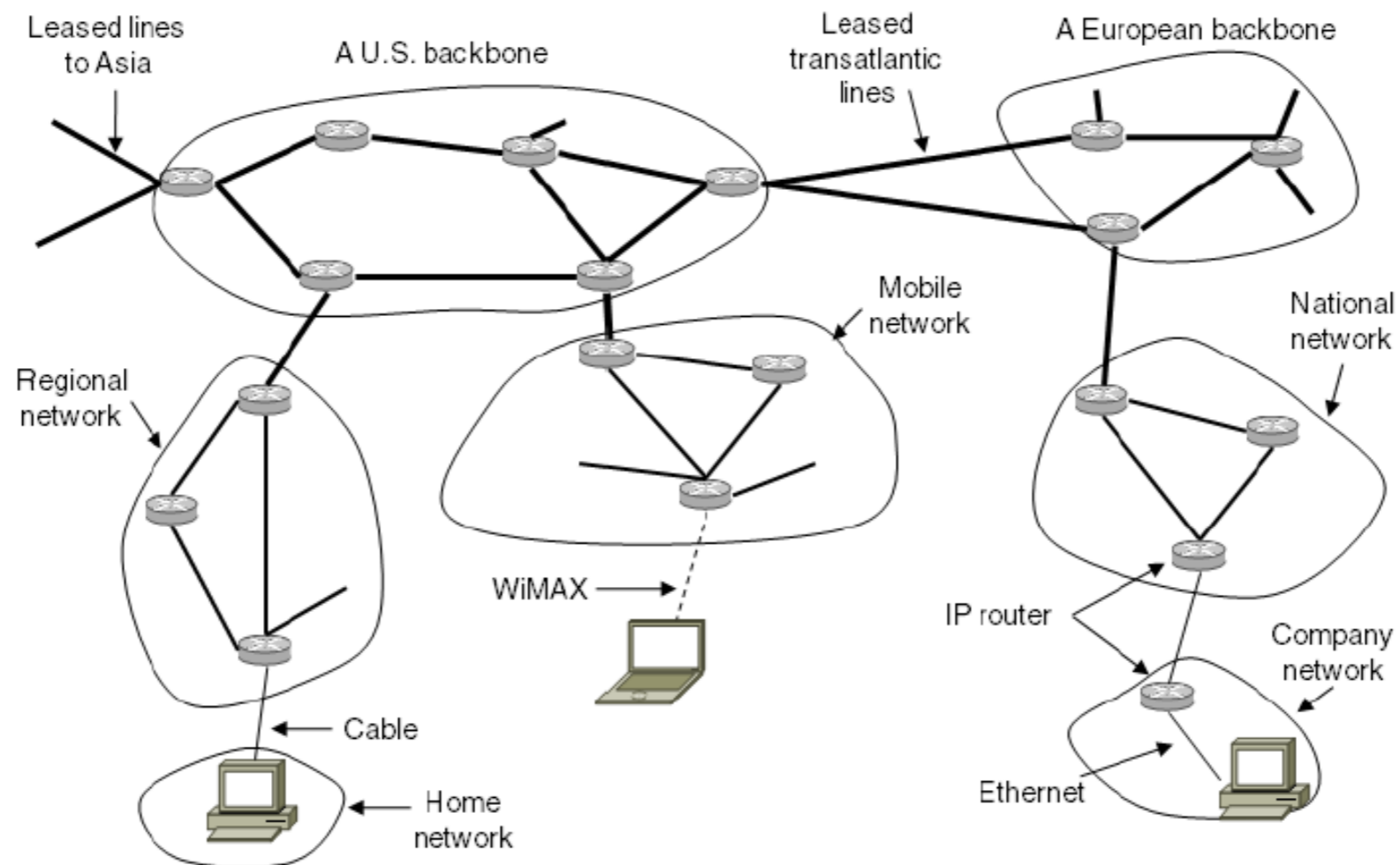
# Network Layer in the Internet

# Internet Protocol

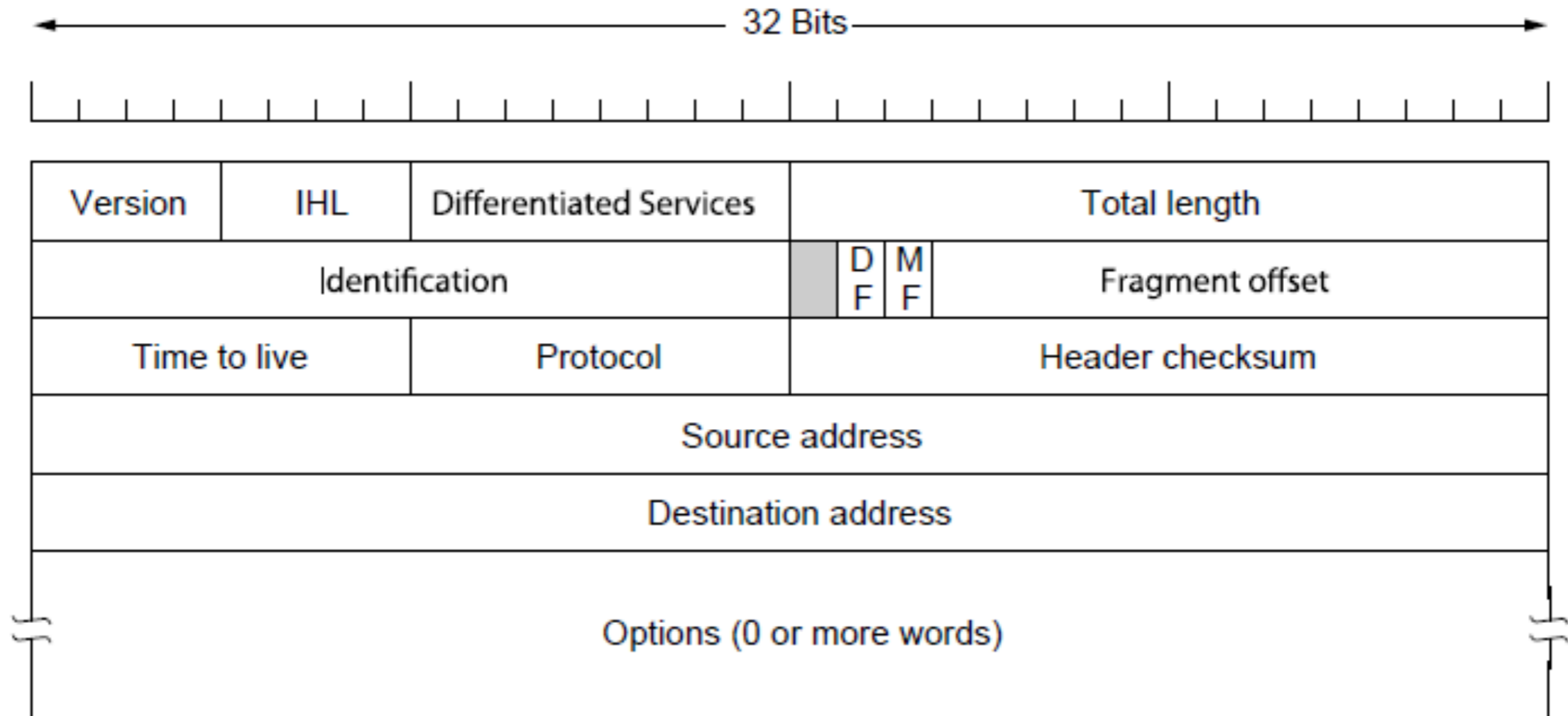
- IP has been shaped by guiding principles:
  - Make sure it works
  - Keep it simple
  - Make clear choices
  - Exploit modularity
  - Expect heterogeneity
  - Avoid static options and parameters
  - Look for good design (not perfect)
  - Strict sending, tolerant receiving
  - Think about scalability
  - Consider performance and cost

# IP

- IP holds many networks together



# IPv4



## IPv4 Header

Big-endian network byte order: The high-order bit of Version goes first.

# IP Addresses

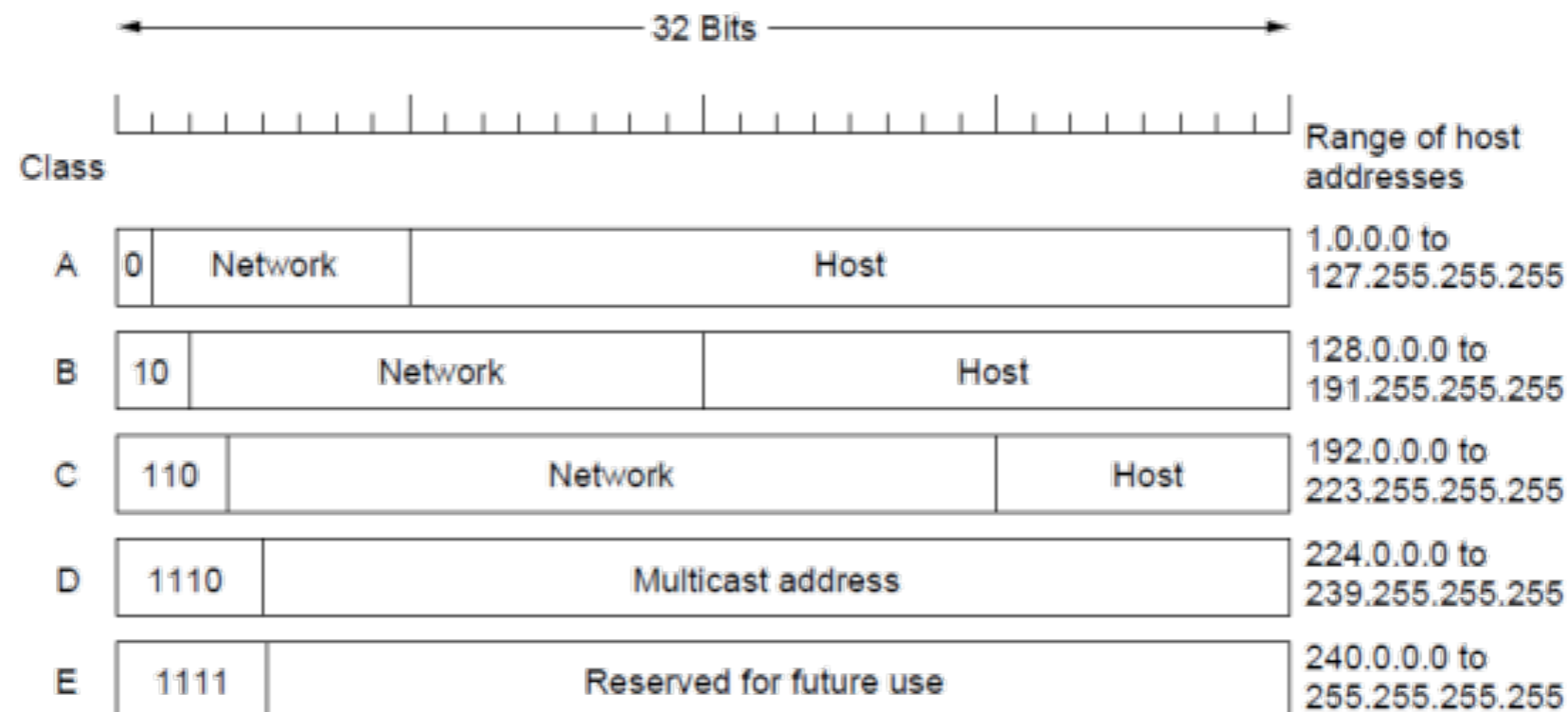
- Split between network and host portion
  - Network portion is the same for all hosts in the same network
- Advantages:
  - Can forward based just on the network portion
- Disadvantages:
  - IP address of a host depends on the network address
    - Impossible to move IP addresses around





# IP-Addresses

- Classful Addressing
  - Old addresses came in blocks of fixed size (A, B, C)
    - Carries size as part of address, but lacks flexibility
    - Called classful (vs. classless) addressing





Internet Assigned Numbers Authority

- IP address ranges controlled by IANA
  - Internet Assigned Number Authority
  - Roots go back to 1972, ARPANET, UCLA
  - Today, part of ICANN
- IANA grants IPs to regional authorities
  - ARIN (American Registry of Internet Numbers) may grant you a range of IPs
- You may then advertise routes to your new IP range
  - There are now secondary markets, auctions, ...

# Class Sizes

Class	Prefix Bits	Network Bits	Number of Classes	Hosts per Class
A	1	7	126 (0 and 127 are reserved)	$2^{24} - 2 = 16,777,214$ (All 0 and all 1 are reserved)
B	2	14	16398	65534 (All 0 and all 1 are reserved)
C	3	21	2097152	254 (All 0 and all 1 are reserved)

- Too many network IDs
- Too many hosts in an A class
- Not enough hosts per C-class

# IP Addresses

- Warning: Class-ful addressing is no longer used
  - Classless Inter-Domain Routing was introduced in 1993

# Classless Inter Domain Routing

- CIDR, pronounced 'cider'
  - Get rid of IP classes
  - Use bitmasks for all levels of routing
  - Aggregation to minimize FIB (forwarding information base)
  - Arbitrary split between network and host
  - Specified as a bitmask or prefix length
    - Example: Northeastern
      - 129.10.0.0 with netmask 255.255.0.0
      - 129.10.0.0 / 16

# Classless Inter Domain Routing

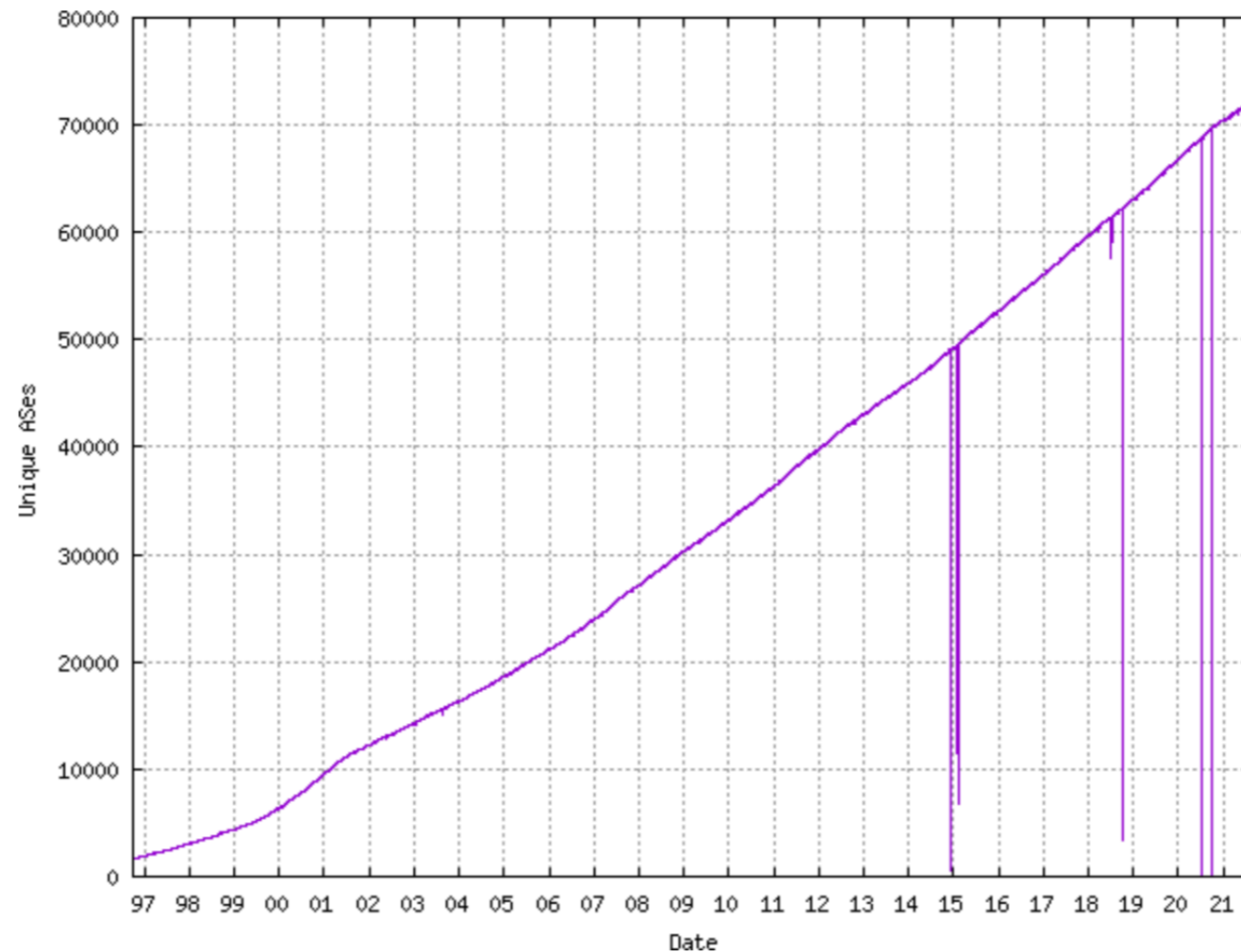
- Original use: aggregating class C ranges
  - One organization given contiguous class C ranges
    - Example: Microsoft, 207.46.192.\* – 207.46.255.\*
    - Represents  $2^{**6} = 64$  class C ranges
- Specified as CIDR address 207.46.192.0/18

207	46	192	0
CF	2E	C0	0
1100 1111	0010 1110	11** ****	**** *****
	Netmask		$2^{**14}$ addresses

# Classless Inter Domain Routing

- [www.cidr-report.org](http://www.cidr-report.org)

Unique AS



Plot Range: 30-Sep-1996 1430 to 28-Oct-2021 1528



# IP-Addresses

- First approach:
- Divide the IP address space into five classes

	Prefix	Network Address	Host ID Length	
<b>Class A</b>	0	7b	24b	
<b>Class B</b>	10	14b	16b	
<b>Class C</b>	110	21b	8b	
<b>Class D</b>	1110	0b	28b	multi-casting
<b>Class E</b>	1111	0b	28b	reserved for research

# IP Addresses

- Example:
  - 134.48.119.146
  - Change to hex: 0x86.0x30.0x77.0x92
  - Change hex to binary:
    - 1000 0110 . 0011 0000 . 0111 0111 . 1001 0010
    - 1000 0110 . 0011 0000 . 0111 0111 . 1001 0010
      - 10 —> Class B
      - Subnet is 00 0110 0011 0000

# IP Addresses

- Classless addressing: 134.48.119.146 → CIDR: 134.48.0.0/16
  - Network address: 134.48.0.0
  - First address: 134.48.0.0
  - Last address: 134.48.255.255

# IP Addresses

- Example:
  - Network range is 5.198.224.0 - 5.198.239.255
  - Network address is 5.198.224.0
    - Last non-zero byte is 224 = 0xe0 = 1110 0000
    - Last non-zero byte is 239 = 0xef = 1110 1111
  - Thus, four bits in this byte plus eight bits in the last byte is 12 bits
  - This gives  $32-12 = 20$
- Thus: network address is 5.198.224.0/20
- (Belongs to an Italian ISP)

# IP Addresses

- What are the first and last addresses of this network:
  - 5.5.42.0 / 23
- Change to hexadecimal:
  - 0x05.0x05.0x2a.0x00
- Change then to binary and count of nine bits from the right ( $32-23 = 9$ )
  - 0000 0101 . 0000 0101 . 0010 1010. 0000 0000
- Red part is the network address, green part is the host address
- Last address is
  - 0000 0101 . 0000 0101 . 0010 1011. 1111 1111
- I.e. 5.5.43.255

# Group Quiz

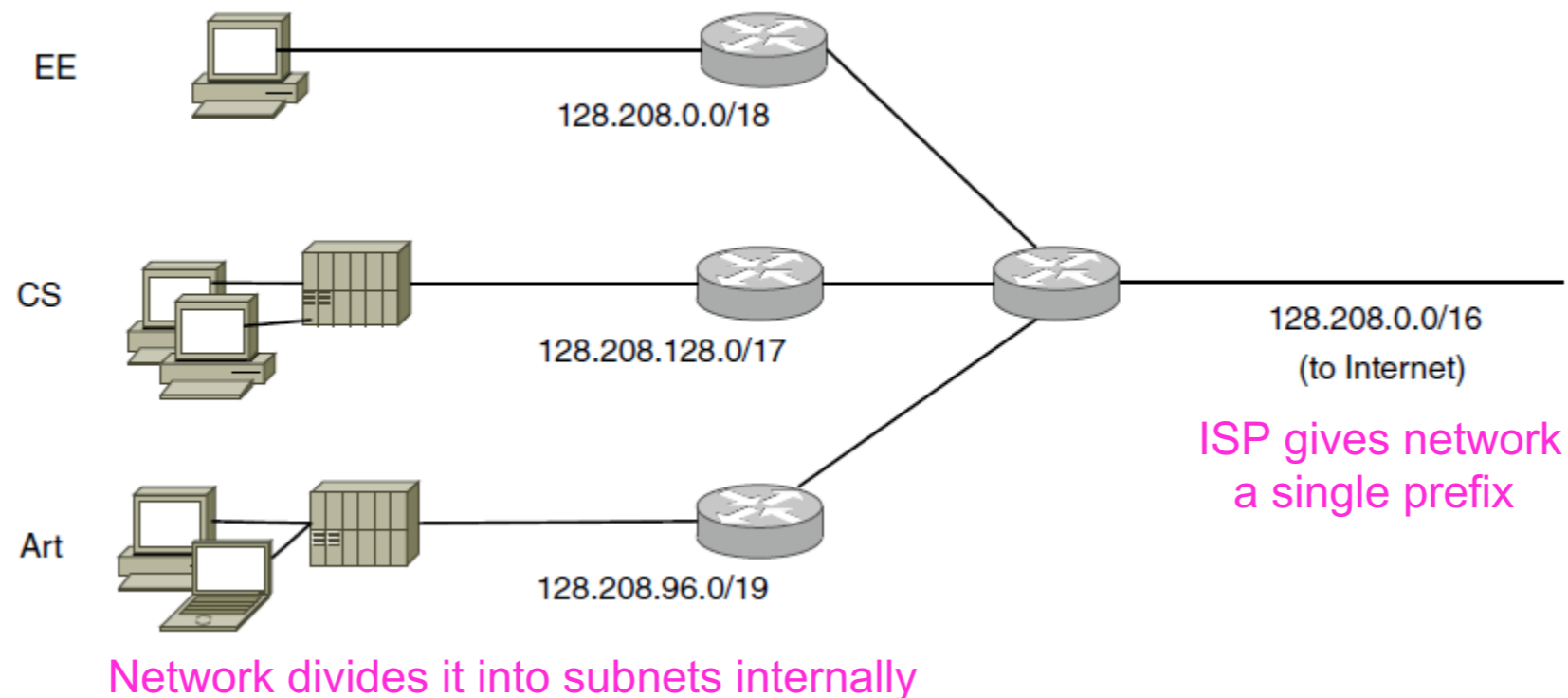
- A classless address is given as 167.199.170.80/27 Find:
  - The number of addresses in the network
  - The network address (prefix)
  - The first address in the network
  - The last address in the network

# Answer

- 167.199.170.64/27
- Subnet is the first 27 bits, i.e. all but the last five bits
- 64 is 0x52 is 0100 0000
  - Network mask is 27 ones followed by 5 zeroes:
    - 0xff . 0xff. 0xff. 0xe0 — 11111111.11111111.11111111.11100000
  - Network address has last byte 0100 0000 -> 64
    - 167.199.170.64
  - First address is 167.199.170.64.
  - Last address has last byte 0101 1111 -> 0x5f -> 95, i.e. is 167.199.170.95
    - Another way is to add  $2^{*5} - 1 = 31$  to the first address

# IP Addresses

- Subnetting
  - splits up IP prefix to help with local management
  - looks like a single prefix outside the network






256 addresses total 64+64+32+32+64	Client 0
	Client 1
	Client 2
	Client 3
	Client 4

Routing Table at ISP	
160.70.14.0 / 26	goes to Client 0 gateway
160.70.14.64/26	goes to Client 1 gateway
160.70.14.128/27	goes to Client 2 gateway
160.70.14.160/27	goes to Client 3 gateway
160.70.14.192/26	goes to Client 4 gateway

Client 0


160.70.14.0/26 -  
160.70.14.63/26



Client 0

Client 1


160.70.14.64/26 -  
160.70.14.127 / 26



Client 1

Client 2


160.70.14.128/27 -  
160.70.14.159/27



Client 2

Client 3


160.70.14.160/27 -  
160.70.14.191/27



Client 3

Client 4

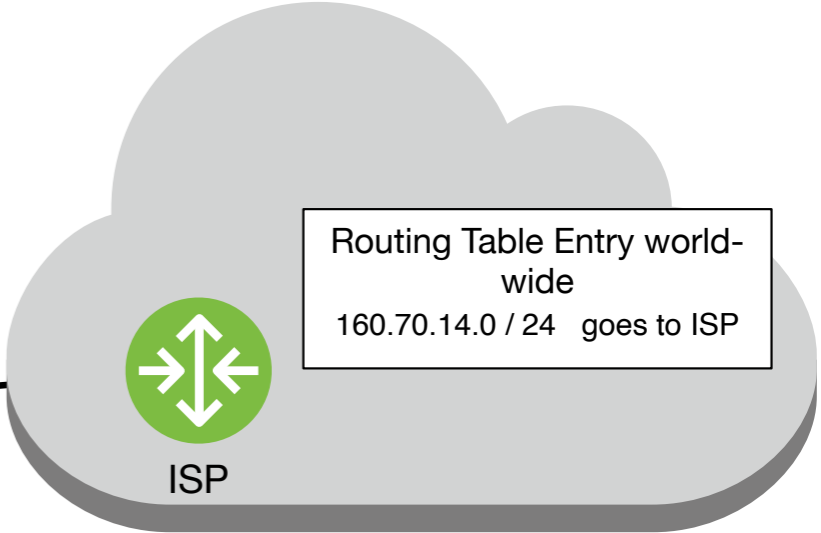
160.70.14.192/26 -  
160.70.14.255/26



Client 4



ISP



ISP

Internet

Routing Table Entry world-wide  
160.70.14.0 / 24 goes to ISP

160.70.14.150

1010 0000 . 0100 0010 . 0000 1110 100 1 0110

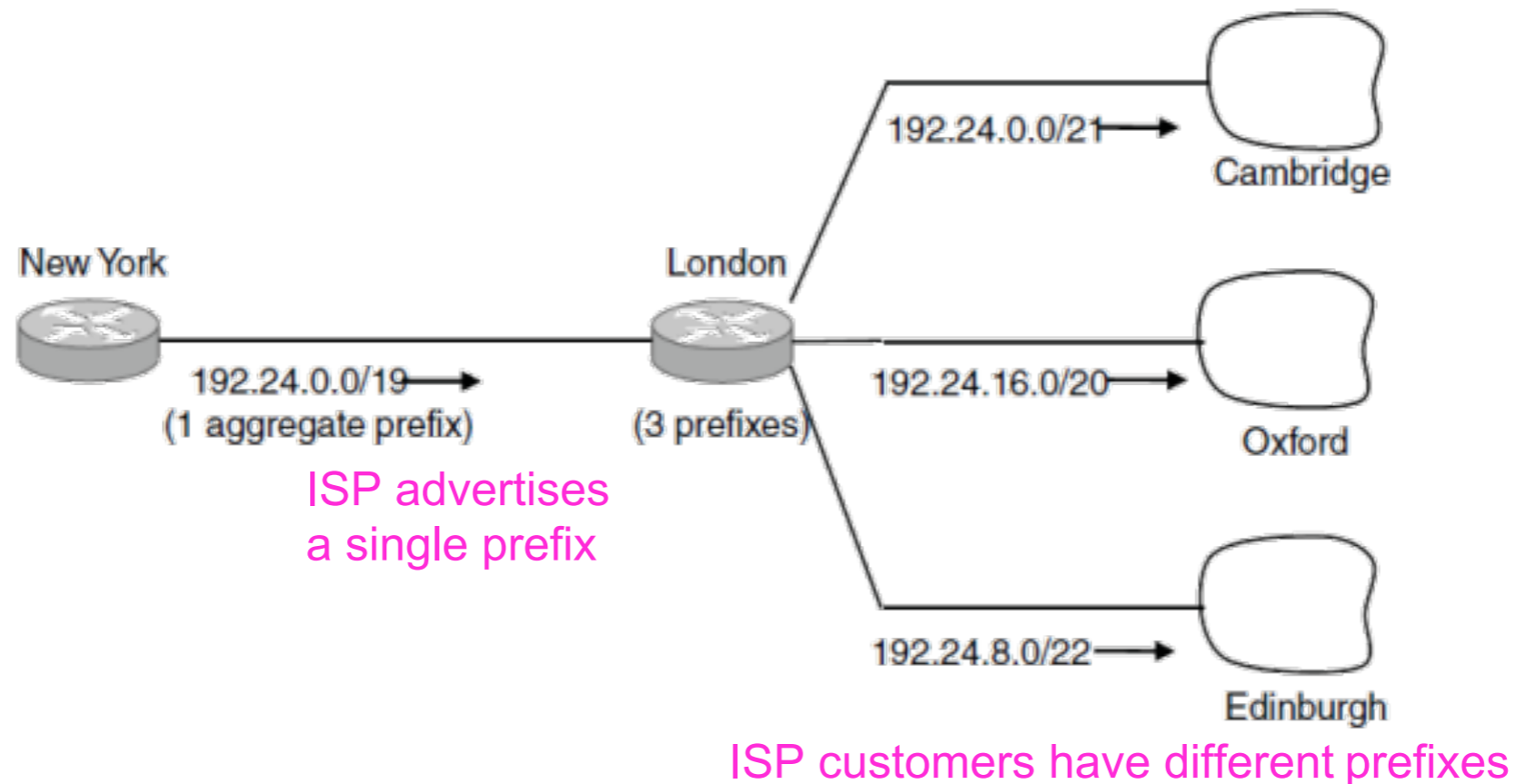
ISP network mask

Stole 3 bits from host address  
for subnetting

5 bits left for host address  
32 addresses left  
(all zero becomes network address,  
all ones becomes broadcast address)

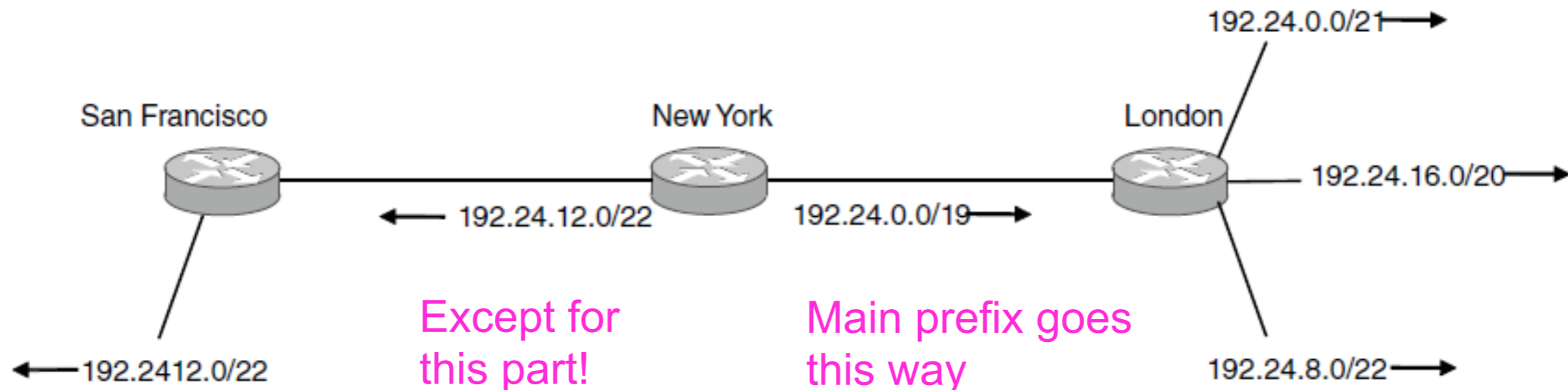
# IP-Addresses

- Aggregation
  - Joins multiple IP prefixes into a single, larger prefix to reduce routing table size



# IP-Addresses

- Packets are forwarded to the entry with the longest matching prefix or smallest address block
  - Complicates forwarding but adds flexibility



# Special IP-Addresses

- This host-address: 0.0.0.0/32
  - Used to get one's own IP address
- Limited broadcast address: 255.255.255.255/32
  - Routers block this address, otherwise it is broadcast
- Loopback address 127.0.0.0/8
  - Packets with an address in this block never leave the host
    - Used for testing
- Private addresses 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 169.254.0.0/16 are not routed
- Multicast addresses 224.0.0.0/4 is reserved for multicasting

# Special IP Addresses

- IPv4 (strongly) recommends:
  - Network address is network mask followed by all zeroes
  - Subnetwork broadcast address is network mask followed by all ones

# Quiz

- A host with IP address 10.240.153.102/26 wants to send a broadcast to all hosts in the subnet.
- Find the broadcast address

# Answer

- Obviously, this is a private address
- Step 1: Calculate the Hex representation of the IP address
  - 10.240.153.102/26
  - 10: 0x0a
  - 240: 0xf0
  - 153: 0x99
  - 102: 0x66

# Answer

- Step 2: Calculate the Hex representation of the IP address
  - 10.240.153.102/26
  - Hex: 0a.f0.99.66 / 26 =  
00001010.11110000.10011001.01100110
  - $26 + 6 = 32$ , therefore the last 6 bits are host address,  
the rest is sub-network address



# Answer

- Step 2: Calculate the Hex representation of the IP address
  - 10.240.153.102/26
  - Hex: 0a.f0.99.66 / 26 =  
00001010.11110000.10011001.01100110
  - Network address:
    - 00001010.11110000.10011001.01**000000**
  - Broadcast address:
    - 00001010.11110000.10011001.01**111111**

# Answer

- Step 3: Convert to decimal
  - 10.240.153.102/26
  - Broadcast address:
    - 00001010.11110000.10011001.01**111111**
  - 10.240.153.127

# FLSM — VLSM

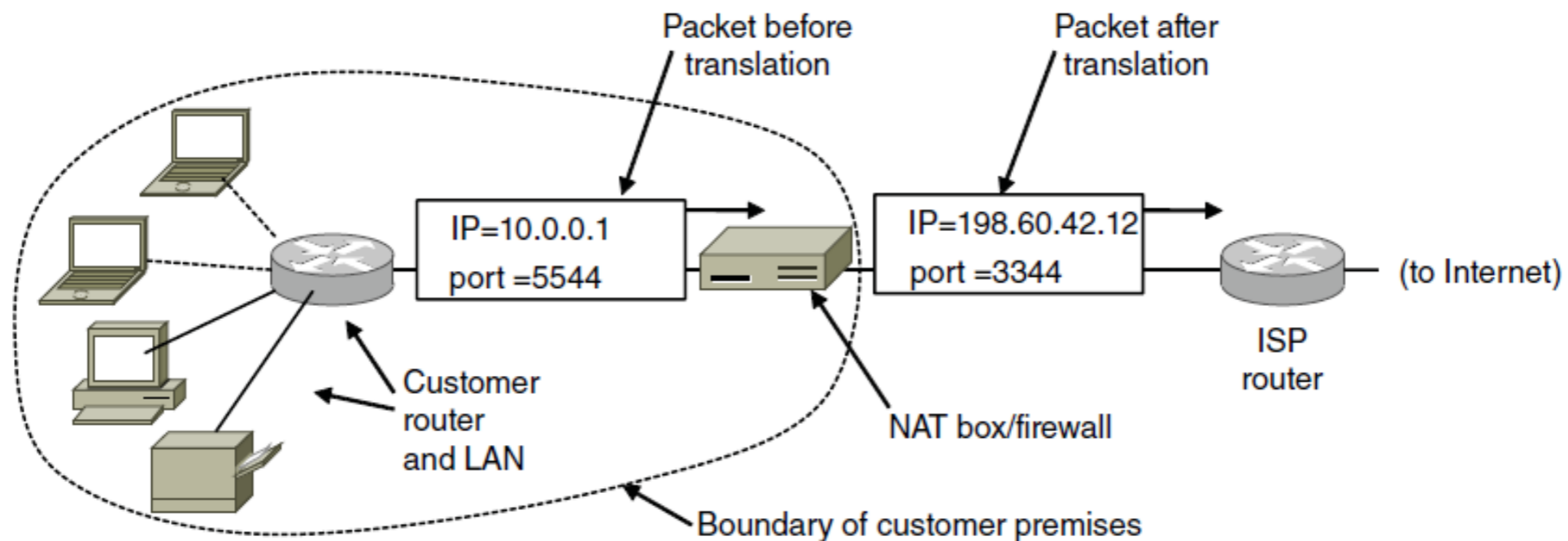
- Classful and CIDR refer to allocation of IP address space by IATA
- FLSM and VLSM refer to allocation within an organization
  - FLSM: Use Fixed Length Subnet Masks
  - VLSM: Use Variable Length Subnet Masks

# FLSM — VLSM

- Example:
  - RIR (Regional Internet Registry) gives you 9.10.11.0 / 24
  - FLSM supports old legacy routing protocols that do not include the subnet mask in advertisements
    - E.g. Need a subnetwork with 29 IP addresses
      - Round up to power of 2:  $2^{5} = 32$
      - $32 - 5 = 27$  bits for subnet-masks
      - 9.10.11.0 / 27, 9.10.11.32/27, 9.10.11.64/27, ...
      - Only  $2^{3} = 8$  subnets
      - Used to be: all zeroes and all ones network not used

# NAT

- NAT (Network Address Translation) box maps one external IP address to many internal IP addresses
  - Uses TCP/UDP port to tell connections apart
  - Violates layering; very common in homes, etc.



# NAT

- Hide a number of hosts behind a single IP address
  - Use reserved ranges as local addresses
    - 10.0.0.0-10.255.255.255
    - 172.16.0.0-172.32.255.255
    - 192.168.0.0-192.168.255.255

# NAT

- Assume that TCP connections are initiated from inside
  - Often enforced by firewalls
  - E.g. 10.5.1.1 makes an http request
- NAT box translates request by replacing the sender address with its address
  - E.g. source address is now 134.48.119.146
  - Make an entry in the transition table:
    - tcp connection to 23.54.50.207 is with 10.5.1.1
  - If there is a reply from 23.54.50.207, NAT replaces the destination from 134.48.119.146 to 10.5.1.1

# NAT

- But this does not always work!
  - This is a request to amazon.com and several local hosts might want a connection with amazon.com
  - NAT is used by ISPs to host many subscribers under the same IP address
- So, NAT uses ports:

- | Private Address | Private Port | External Address | External Port | Protocol |
|-----------------|--------------|------------------|---------------|----------|
| 10.5.1.1        | 1400         | 23.54.50.20      | 80            | TCP      |
| 10.5.1.15       | 1401         | 23.54.50.20      | 80            | TCP      |
| ...             | ...          | ...              | ...           | ...      |

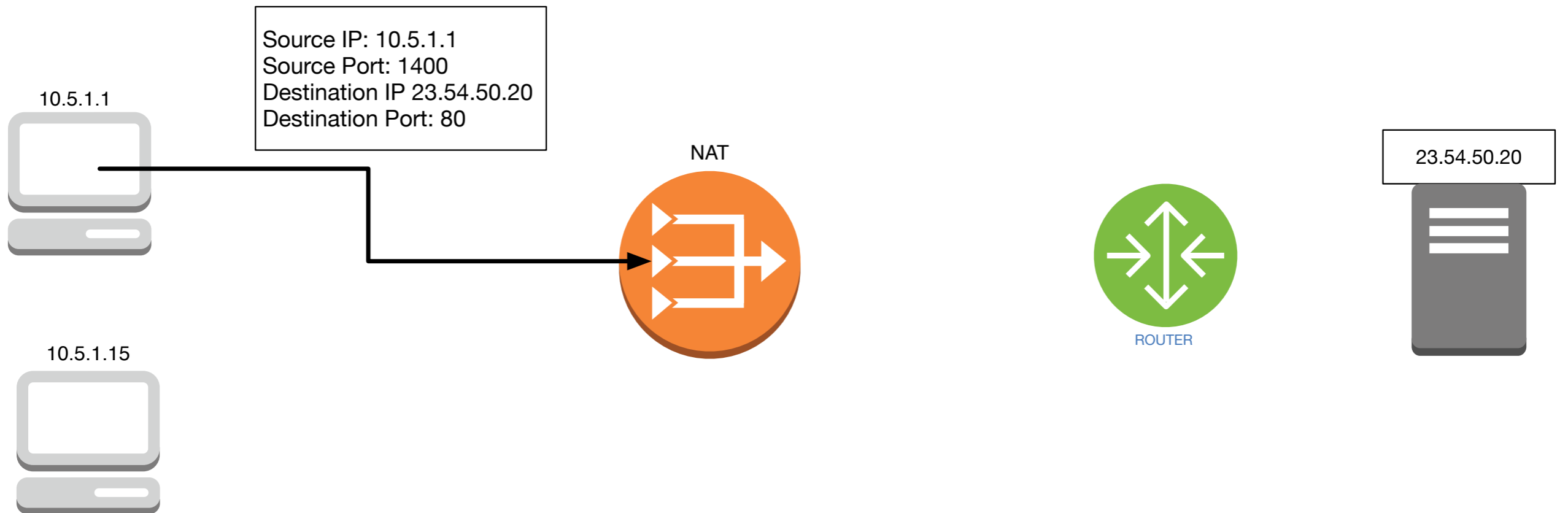


# NAT

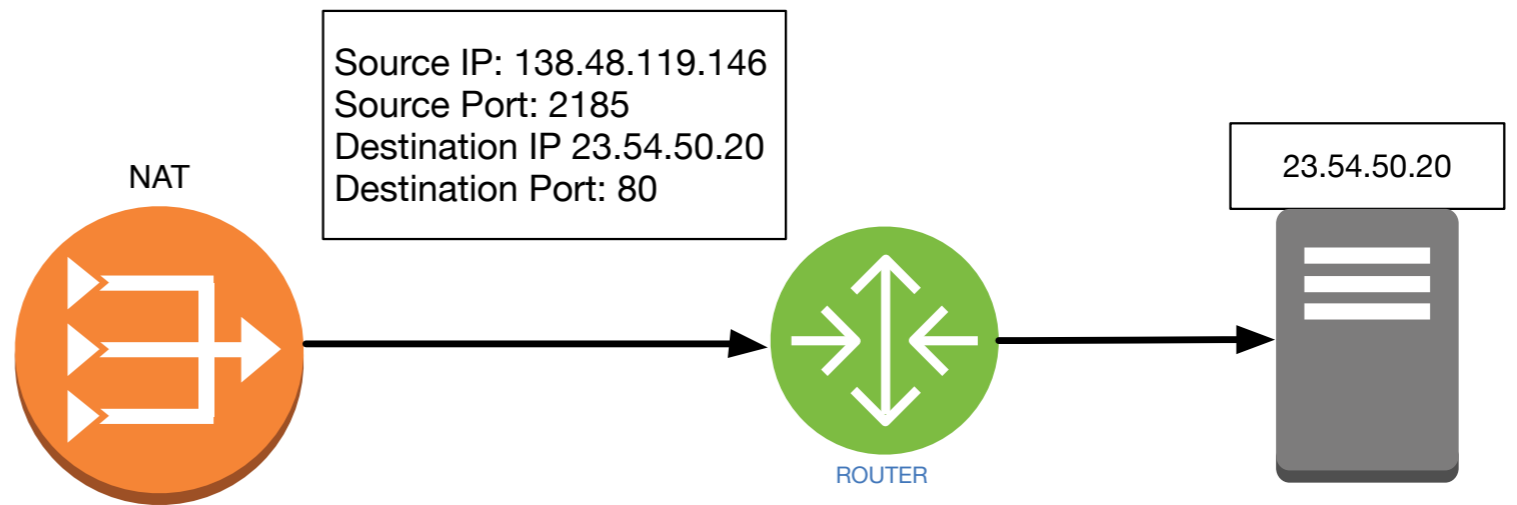
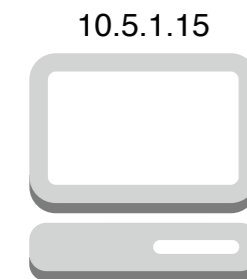
- When a TCP packet arrives to 134.48.119.146 at port 1400:
  - Look up NAT table
  - Replace destination address with 10.5.1.1
  - Replace

Private Address	Private Port	External Address	External Port	Protocol
10.5.1.1	1400	23.54.50.20	80	TCP
10.5.1.15	1401	23.54.50.20	80	TCP
...	...	...	...	...

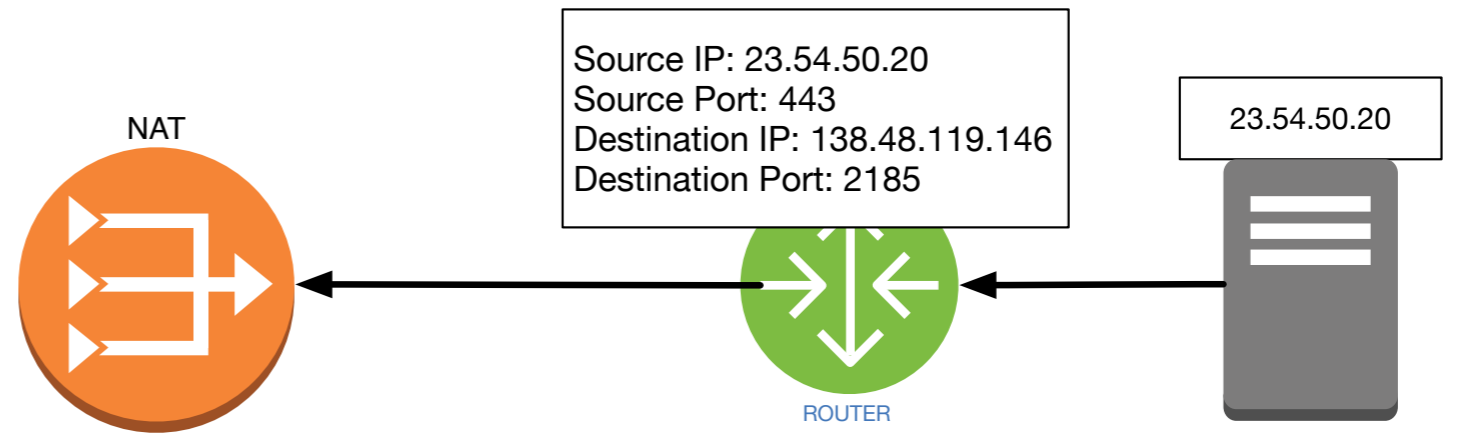
# NAT



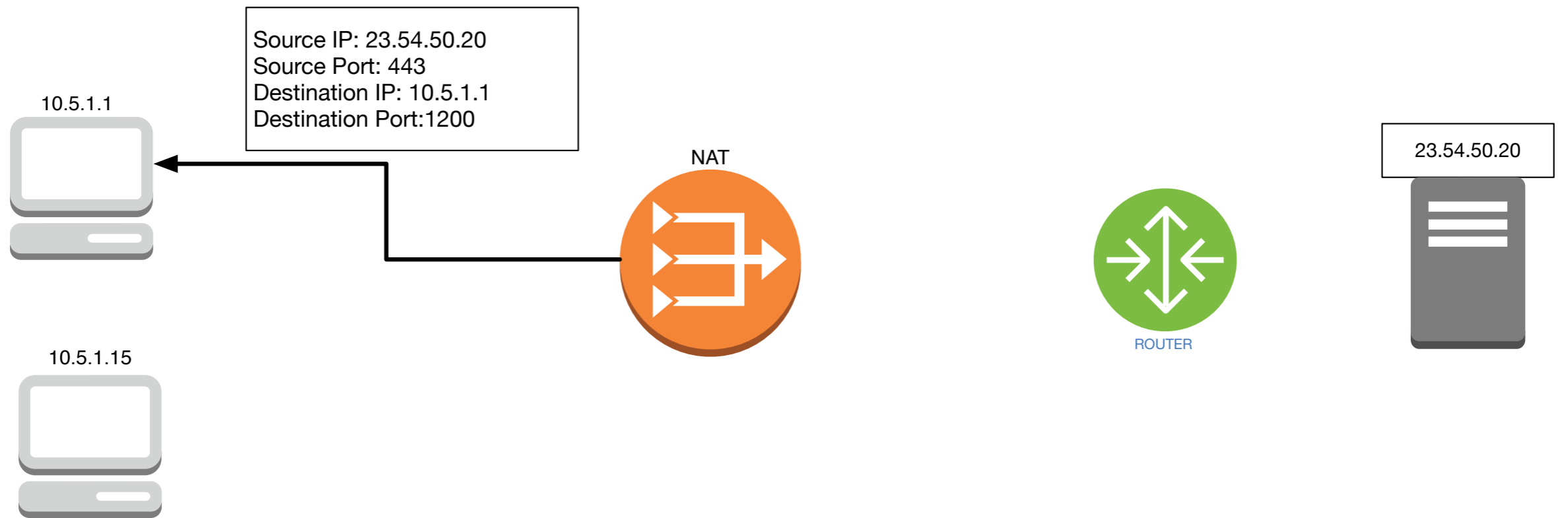
# NAT



# NAT



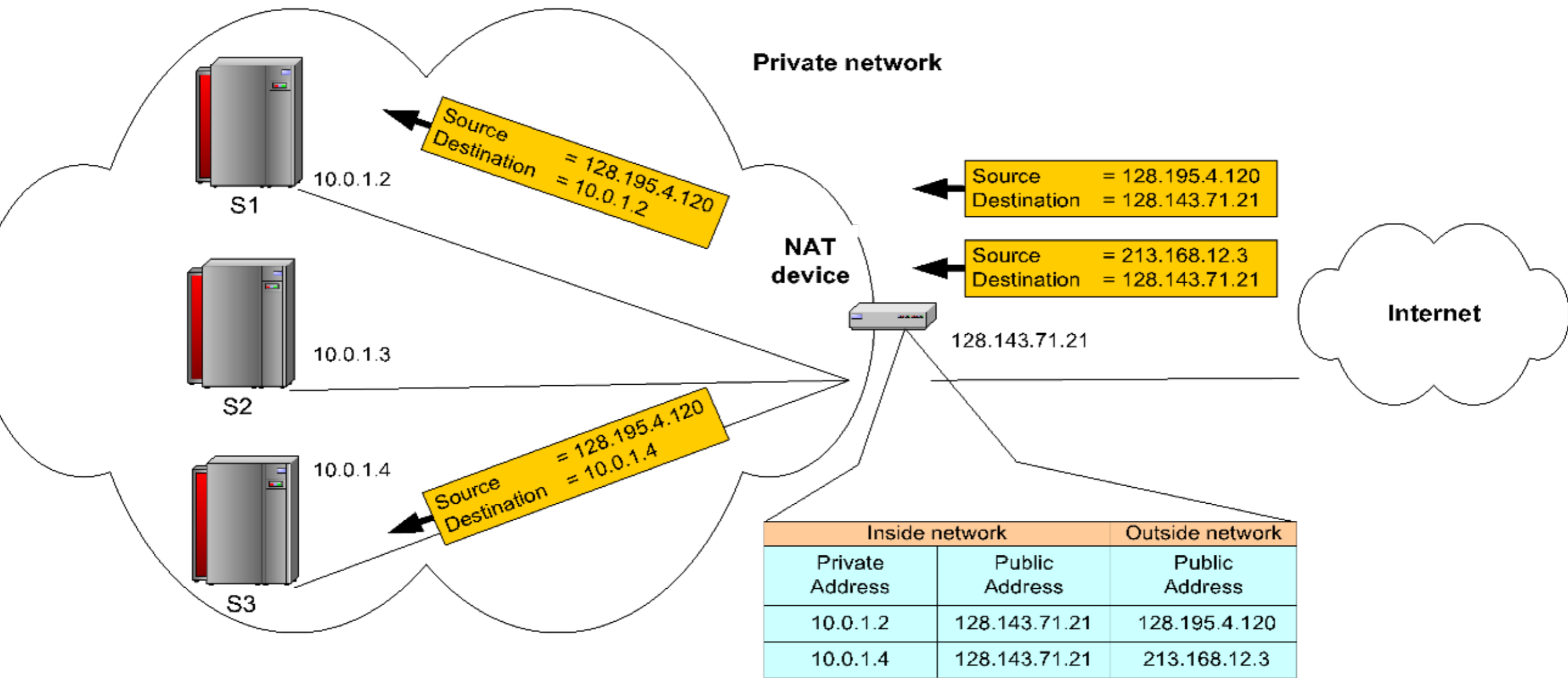
# NAT



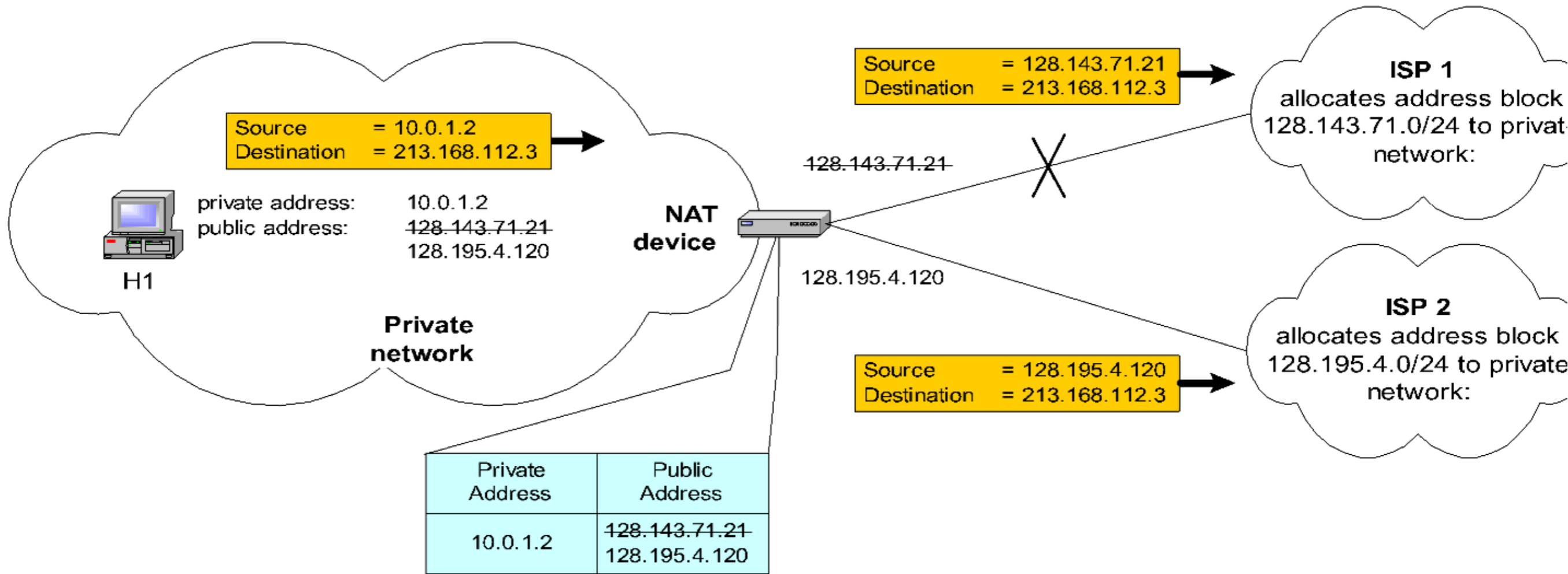
# NAT

- NAT disables a number of protocols
- NAT can be used for load balancing among servers
- NAT can be used for multi-homed networks
  - A company has contracted internet access from two alternative hosts

# NAT Load Balancing



# NAT Multi-Hosting



The same computer is accessible via two different addresses



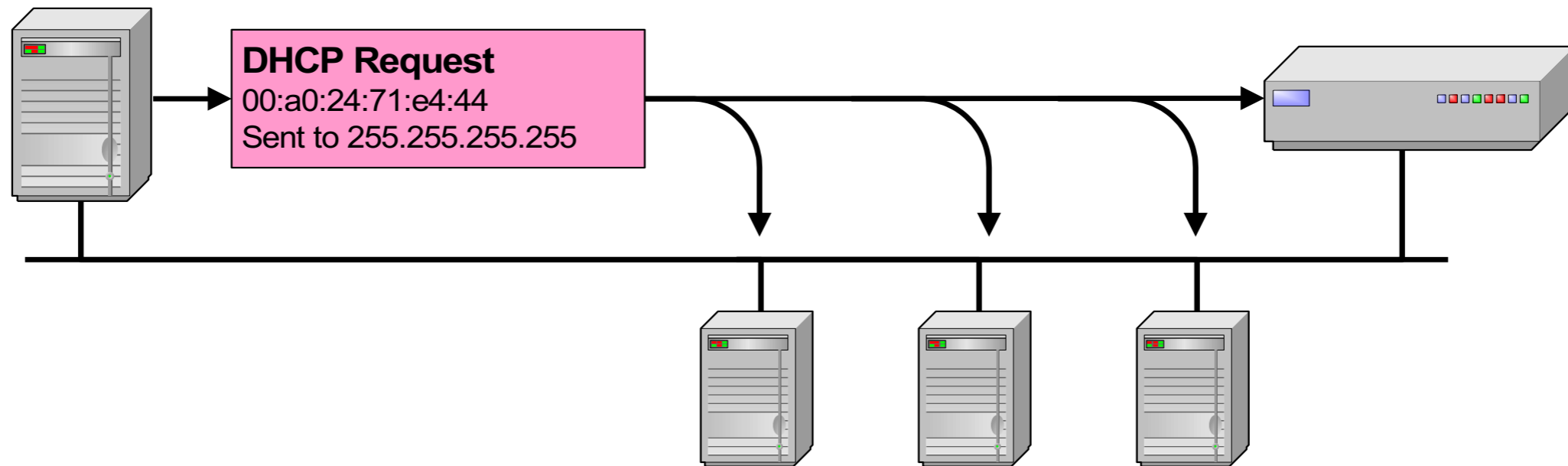
# DHCP

- Dynamic Host Configuration Protocol
  - Provides IP-address, network prefix, default gateway, name server address
- Host creates a DHCP-discover message in UDP with source address 0.0.0.0 and destination address 255.255.255.255 and ephemeral ports 68 and 67
- DHCP server(s) responds with DHCP-offer message
- Host selects an offer and responds with a DHCP-request message to the offering server
- DHCP server answers with a DHCP-ack

# DHCP

Argon  
00:a0:24:71:e4:44

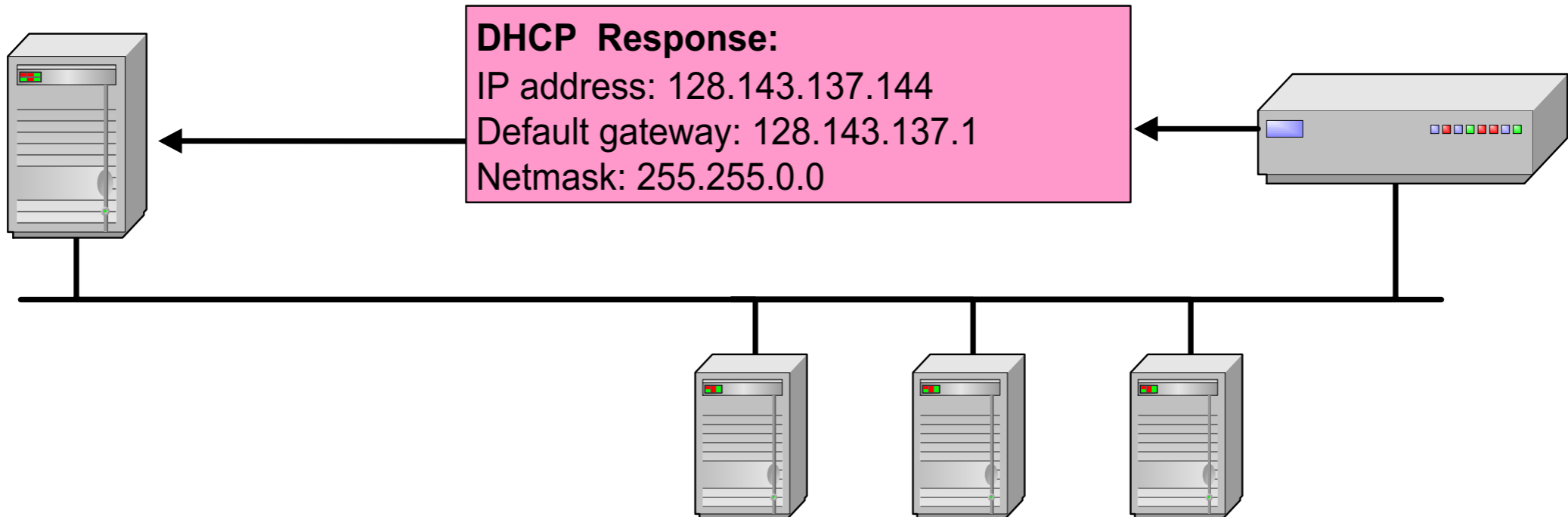
DHCP Server



# DHCP

Argon  
**128.143.137.144**  
00:a0:24:71:e4:44

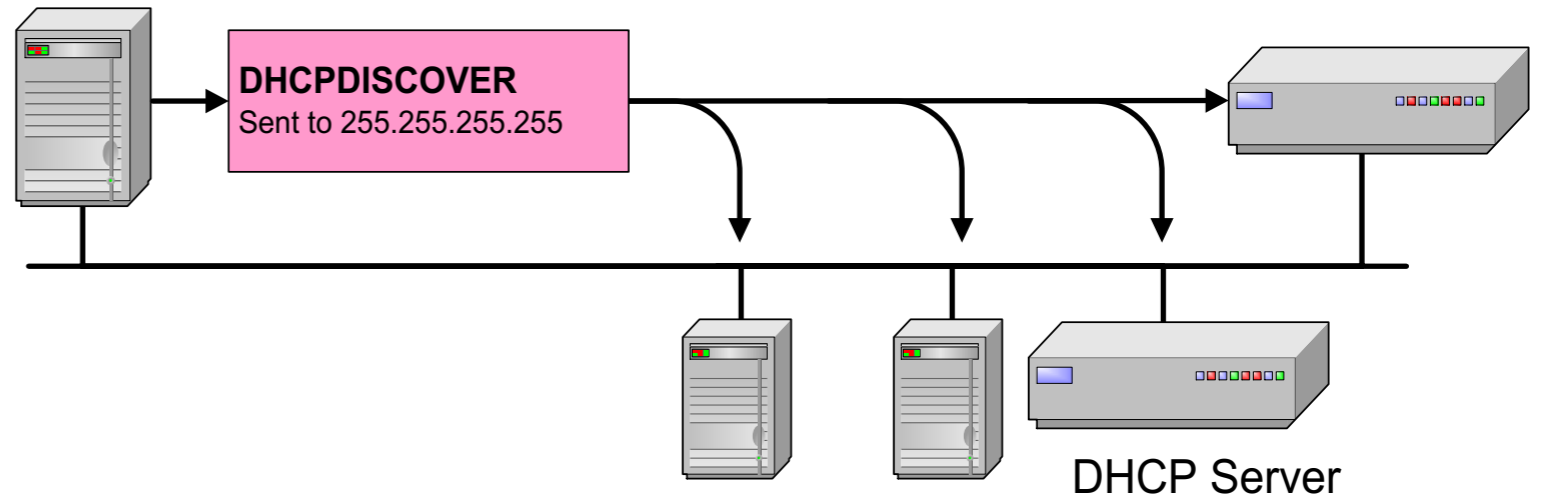
DHCP Server



# DHCP

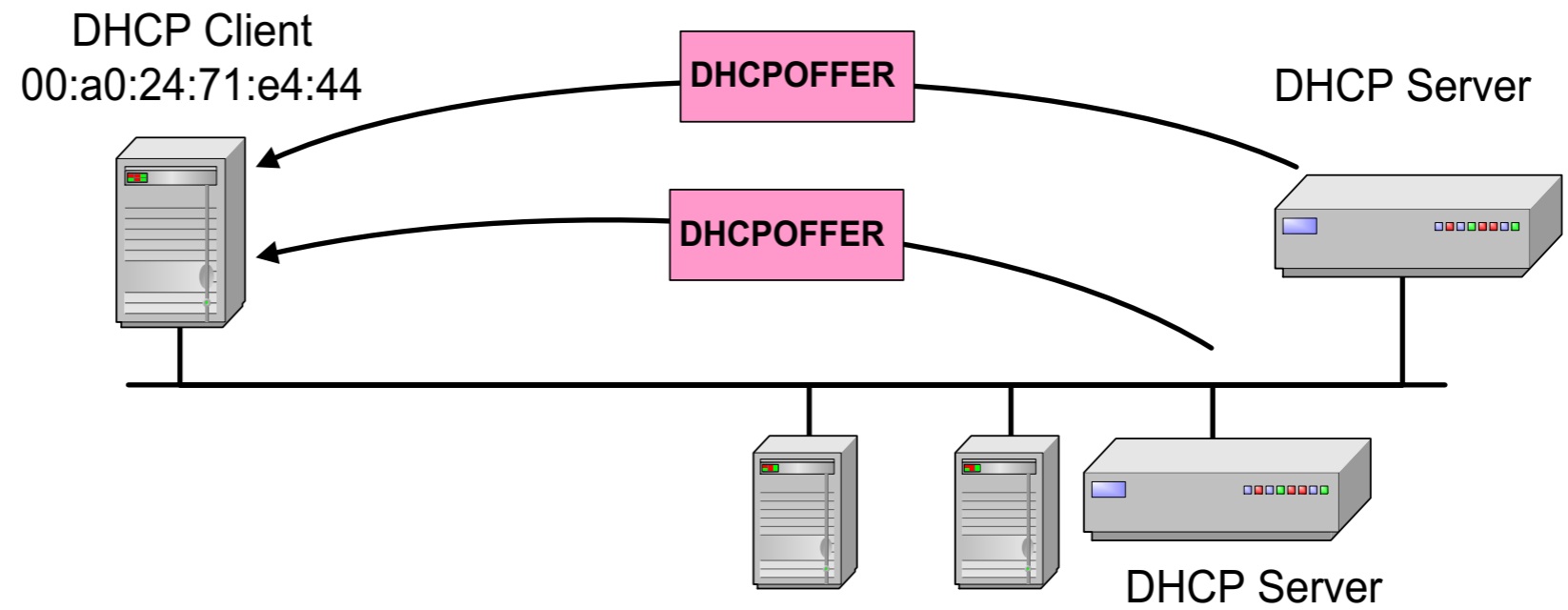
## DHCP Discover

DHCP Client  
00:a0:24:71:e4:44



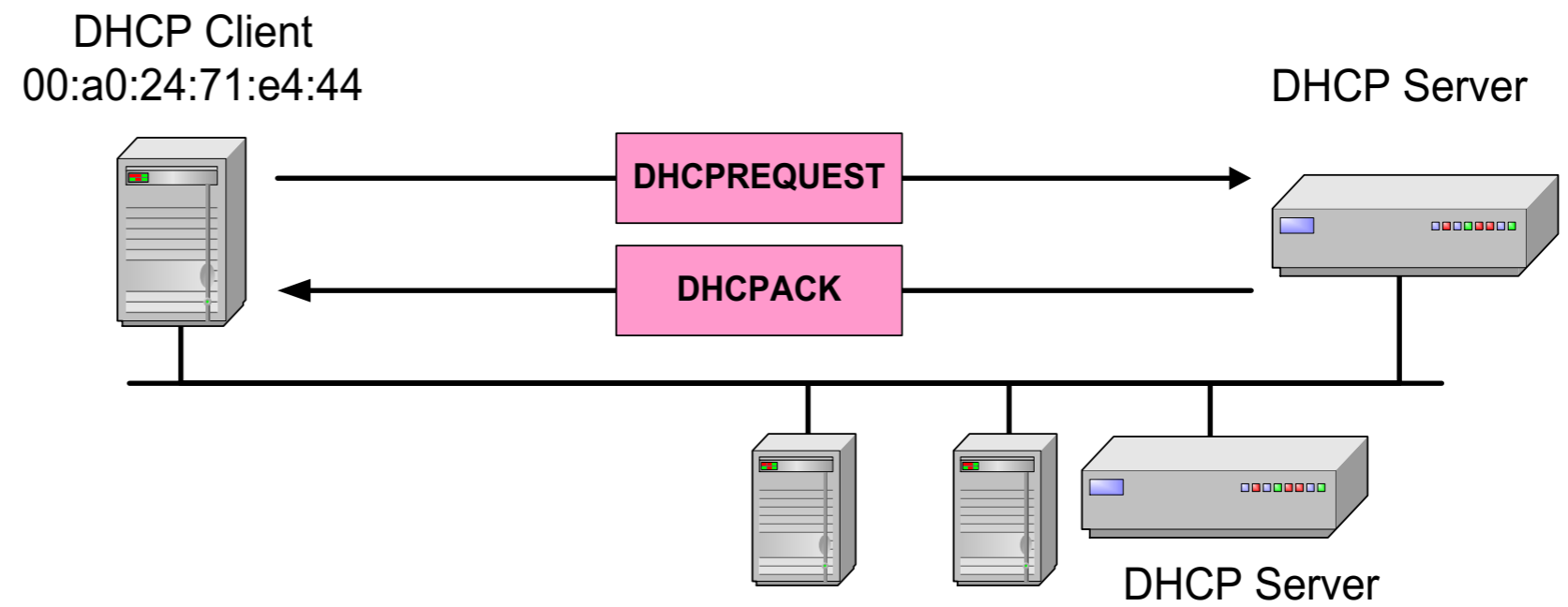
# DHCP

## DHCP Offer



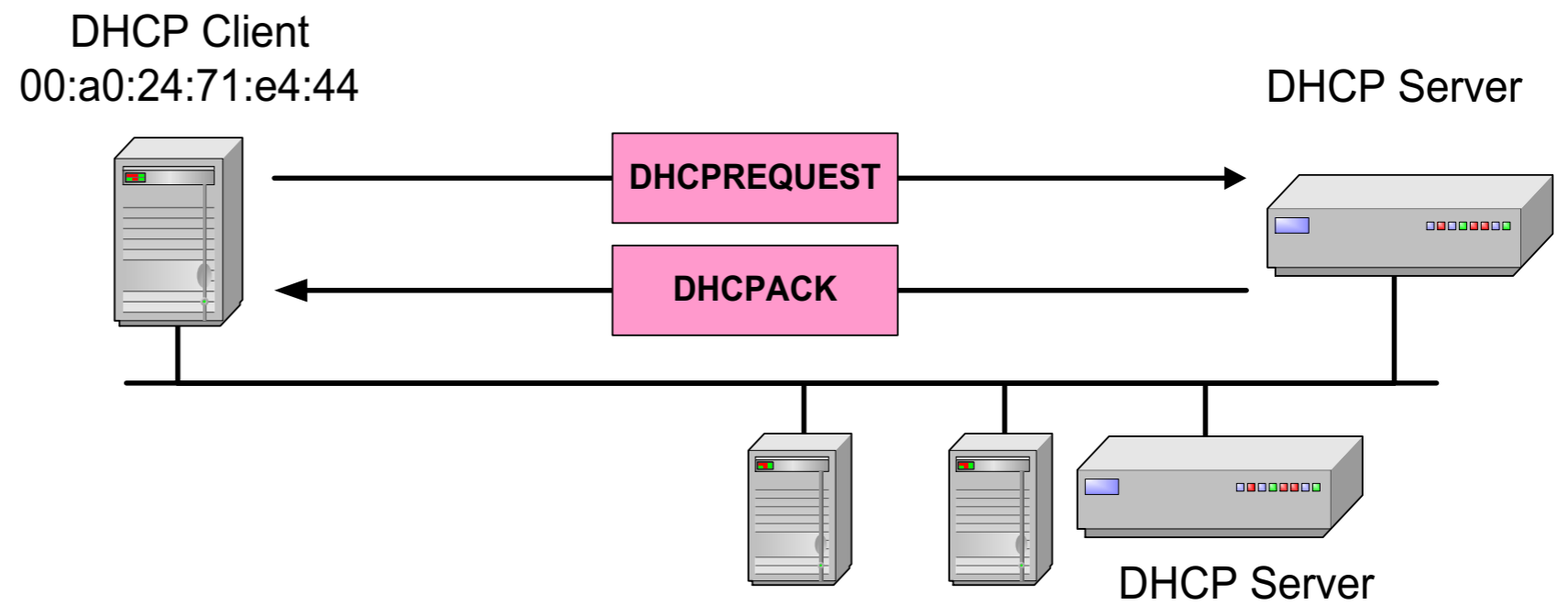
# DHCP

## DHCP Discover



# DHCP

## DHCP Lease Renewal

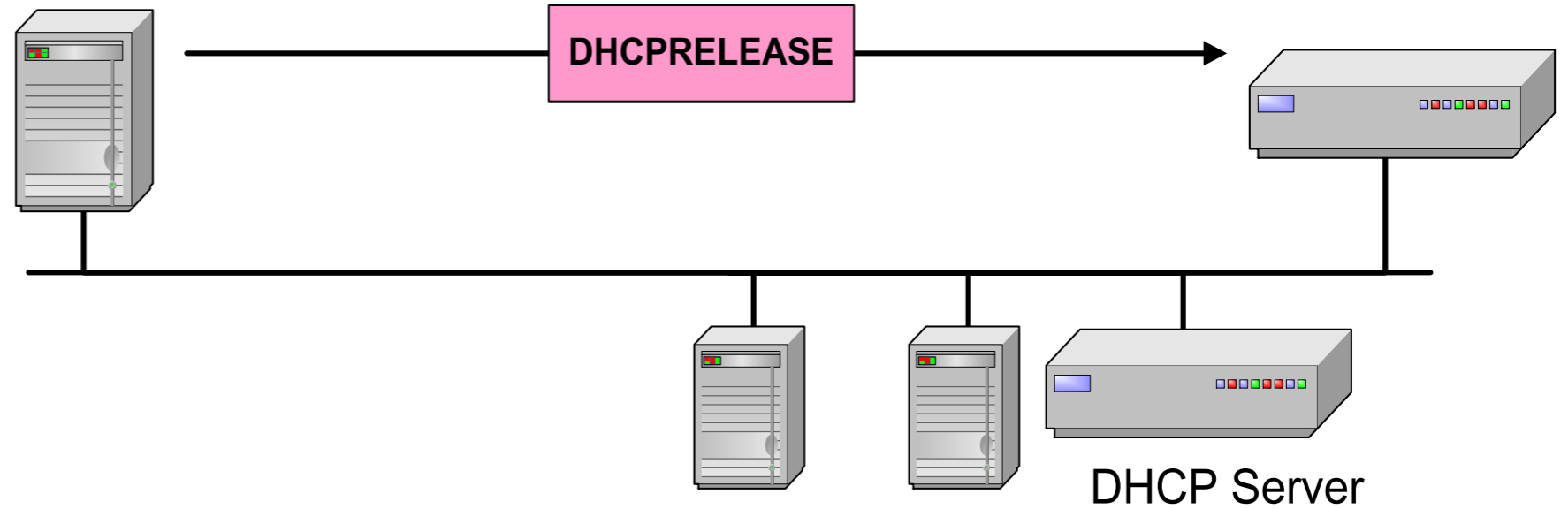


# DHCP

## DHCP Release

DHCP Client  
00:a0:24:71:e4:44

DHCP Server

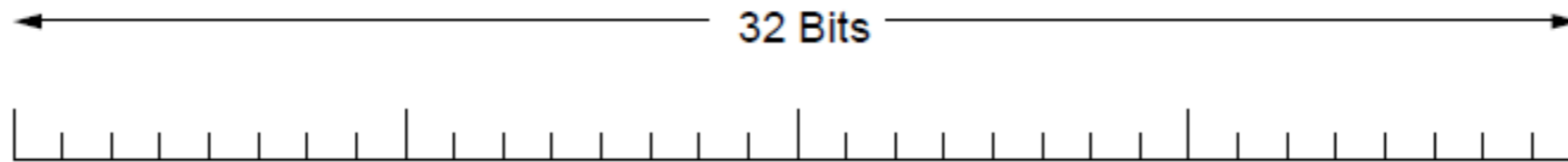




# IP v6

- Major upgrade in the 1990s due to impending address exhaustion, with various other goals:
  - Support billions of hosts
  - Reduce routing table size
  - Simplify protocol
  - Better security
  - Attention to type of service
  - Aid multicasting
  - Roaming host without changing address
  - Allow future protocol evolution
  - Permit coexistence of old, new protocols, ...
- Deployment has been slow & painful, but may pick up pace now that addresses are all but exhausted

# IPv6



Version	Diff. Serv.	Flow label	
Payload length		Next header	Hop limit
Source address (16 bytes)			
Destination address (16 bytes)			

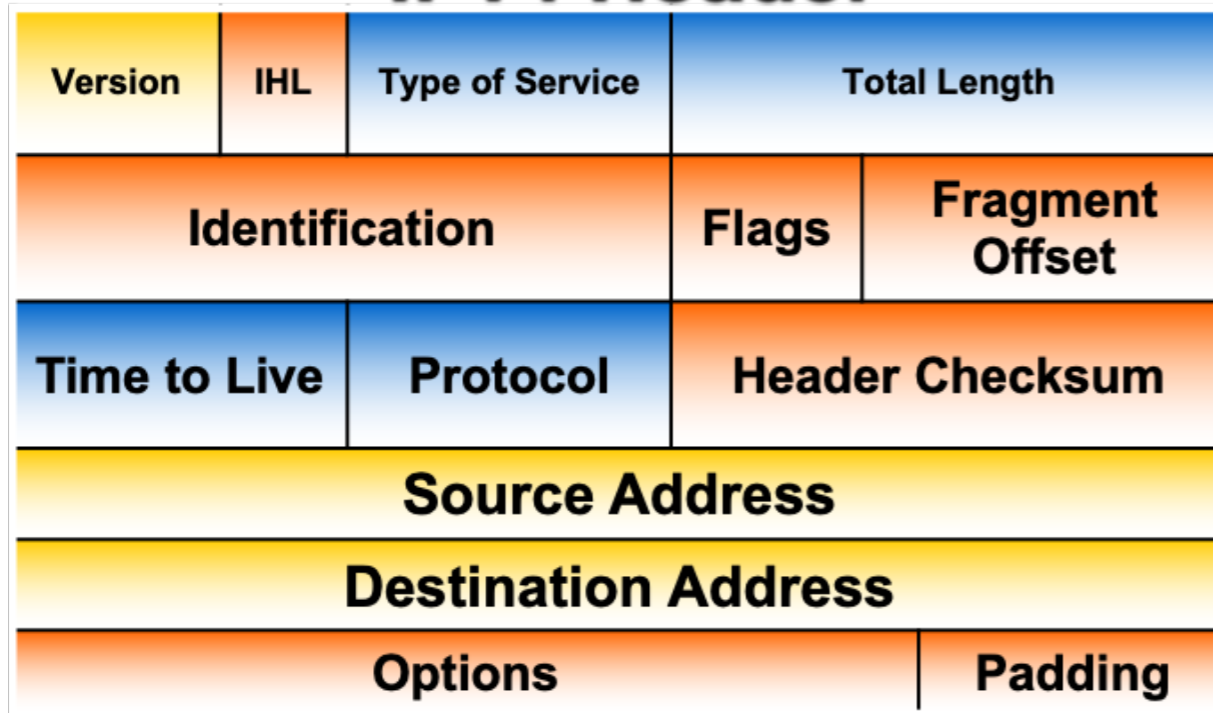
# IPv6

- IPv6 extension headers handle other functionality

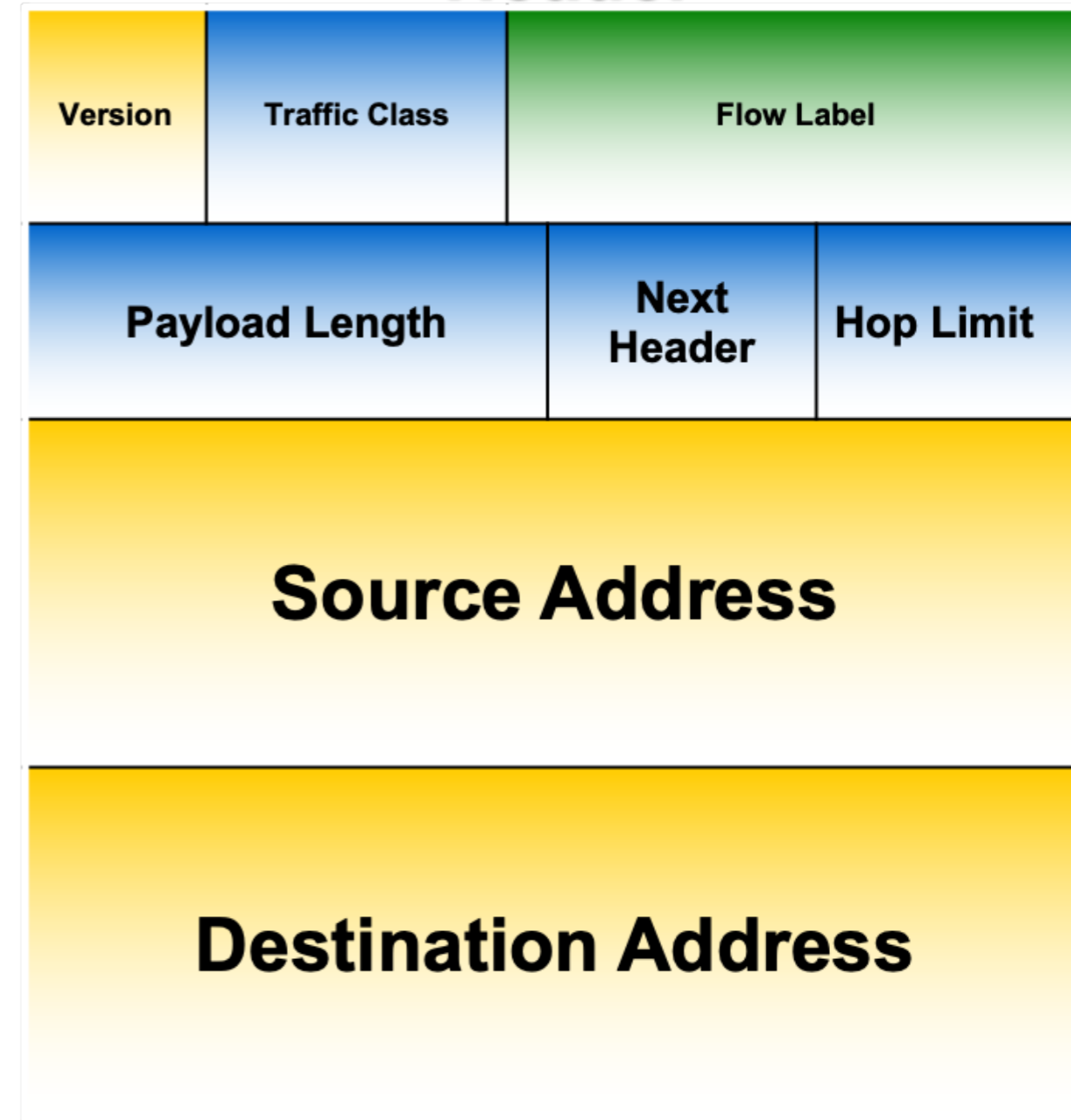
<b>Extension header</b>	<b>Description</b>
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents





# IPv4 vs IPv6

## IPv4 Header



## IPv6 Header



-  - field kept from IPv4 to IPv6
-  - field not kept in IPv6
-  - name and position changed in IPv6
-  - new field in IPv6

# IPv4 vs IPv6

- Streamlined
  - Fragmentation fields moved out of base header
  - IP options moved out of base header
  - Header Checksum eliminated
  - Header Length field eliminated
  - Length field excludes IPv6 header
  - Alignment changed from 32 to 64 bits

# IPv4 vs IPv6

- Revised
  - Time to Live ' Hop Limit
  - Protocol ' Next Header
  - Precedence & TOS ' Traffic Class
  - Addresses increased 32 bits ' 128 bits
- Extended
  - Flow Label field added

# IPv4 vs IPv6

- Routers no longer fragment
  - Instead: send ICMP “packet too big” to source
- Fragmentation is now “end-to-end”
- However: Legal to use IPv6 extension fragment header

# IPv4 vs IPv6

- Addresses: 8 hexadecimal numbers from 0 to 0xFFFF separated by colons
- String of zeroes can be replaced by ::
  - FFA0:1234:0:0:0:0:16:A03 becomes FFA0:1234::16:A03
- IPv4 mapped addresses:
  - 0:0:0:0:0:FFFF:10.1.68.3



# Group Activities

- What is the full address of
  - 3210:a0::4567:89ab

# Answer

- What is the full address of
  - 3210:a0::4567:89ab
  - 3210:00a0:0000:0000:0000:0000:4567:89ab

# Group Activities

- Which is the shortest representation of
  - 0123:0248:0000:0098:abce:0000:0000:da98

# Answer

- Which is the shortest representation of
  - 0123:0248:0000:0098:abce:0000:0000:da98
- We can only use double colons to compress one of the strings of hex-0 and pick the larger one
  - 123:248:0:98:abce::da98

# IPv6

- Unicast addresses
  - defines a single interface (router or computer or device)
  - LSB are derived from Mac address
  - Addresses 2000/3
  - Global routing prefix + Subnet Identifier + Interface identifier
- Anycast addresses
  - Group of computers that share a single address
  - Message sent to one of the computers
- Multicast addresses: FF00/8
- NO broadcasting (other than multicasting)

# IPv6

- Autoconfiguration
  - When joining a network:
    - Calculate local link address as link-local prefix followed by zeroes followed by 64b interface id from network card
    - Test that local link address is unique
    - Sends router solicitation message
    - Wait for a router advertisement message
      - which gives the global unicast prefix plus subnet prefix

# IPv6 Subnetting

- IANA "owns" the entire IPv6 address space
- Uses the 2000::/3 prefix for global unicast addresses
  - All IPv6 addresses starting with 001\*
- Assigns address blocks to regional authorities
  - E.g. LACNIC got 2001:1200::/23

# Activity

- Calculate the network prefix in binary for LACNIC
  - 2001:1200::/23
  - **0010 0000 0000 0001:0001 001**0 0000 0000
  - All but the last bit of the first six bytes



# IPv6 Subnetting

- ISP acquire a range of blocks from their regional authority
  - Example:
    - 2800:a0::/28 for Administración Nacional de Telecomunicaciones, Uruguay
    - (28/4 = 7: everything starting with 2800:00a.)
      - Remember that lacking digits are padded with zeroes at the beginning

# IPv6 Subnetting

- ISP then assigns address ranges to customers
  - E.g. 2800:a0:10:503::/64
    - Written out: 2800:00a0:0010:0503::
- Typical assignments:
  - ISP: 32 bits, Customers 48, 64 bits prefix length

# Group Activity

- What is the network prefix of this IPv6 address:
  - 2605:a000:45c4:de00:65be:10cd:6fb3:e114/48
  -

# Answer

- What is the network prefix of this IPv6 address:
  - 2605:a000:45c4:de00:65be:10cd:6fb3:e114/48
- $48/4 = 12$ : Use the first 12 hex digits
  - 2605:a000:45c4::/48

# IPv6 Broadcasting

- The broadcasting capability is subsumed by multicasting
  - IPv6 multicasting uses prefix ff00::/8
    - Start out with 8 one-bits
    - Other fields in the first three bytes are used for flags and scope
- IANA set aside certain multicast addresses
  - E.g. ff02::1 for all nodes on the local network
  - ff02::2 for all routers on the local network
  - ff02::5 All SPF routers in OSPFv3

# Internet Control Protocols

# Internet Control Protocols

- IP works with the help of several control protocols:
  - ICMP is a companion to IP that returns error info
    - Required, and used in many ways, e.g., for traceroute
  - ARP finds Ethernet address of a local IP address
    - Glue that is needed to send any IP packets
    - Host queries an address and the owner replies
  - DHCP assigns a local IP address to a host
    - Gets host started by automatically configuring it
    - Host sends request to server, which grants a lease

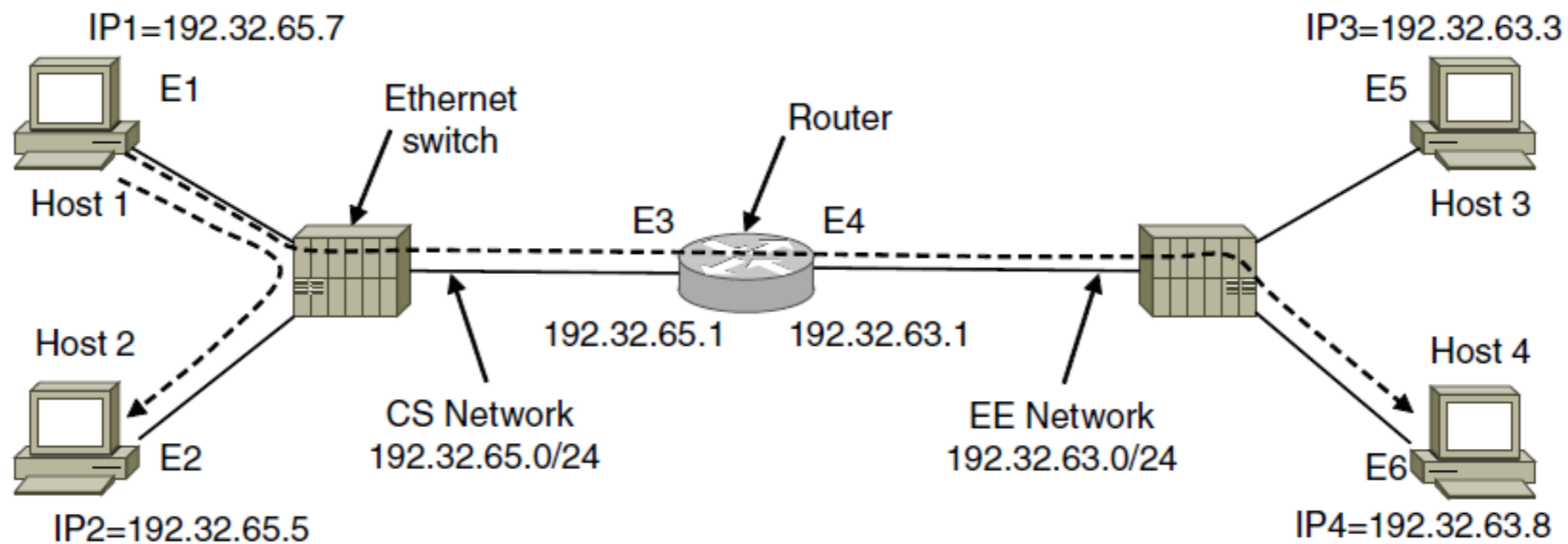
# Main ICMP types

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and Echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router



# Address Resolution Protocol (ARP)

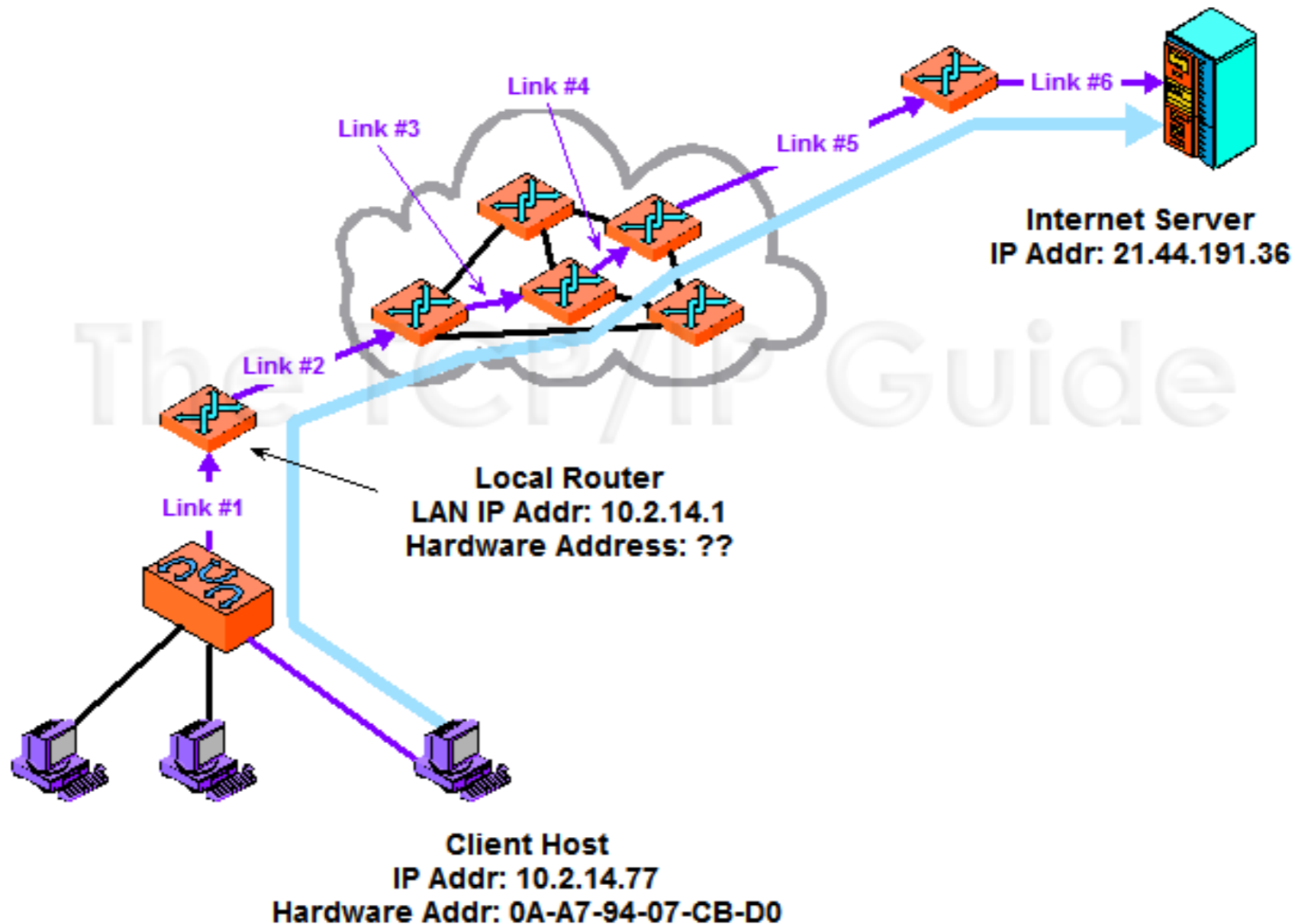
- Lets nodes find target Ethernet addresses from their IP addresses



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

# Address Resolution Protocol (ARP)

- Client knows the IP (layer 3) address of Internet Server
- Package needs to be routed at layer 2

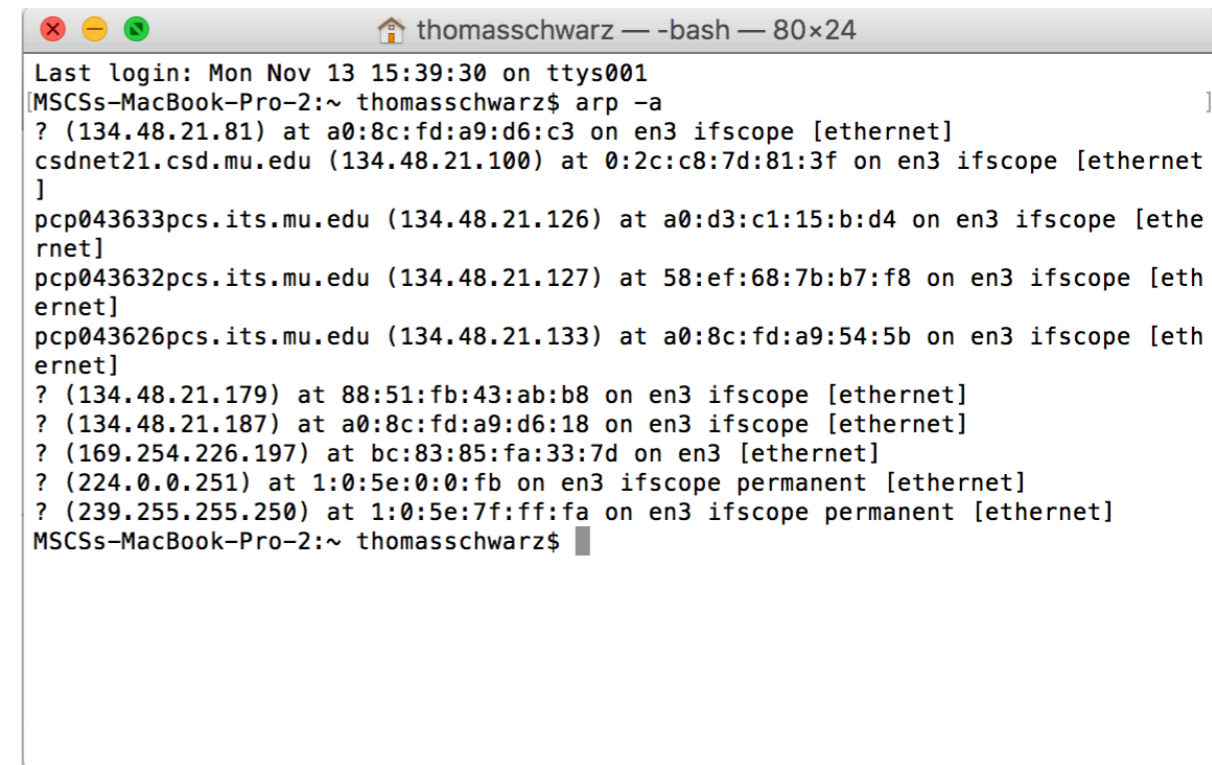


# ARP: Dynamic Address Resolution

- Device that wants to send broadcasts an ARP request
  - 87:32:10:50:FF:34: Who-has-128.32.1.254
- Routers will not route this type of package
- Switches will broadcast
- Device with IP address 128.32.1.254 will respond directly to initiator
  - 128.32.1.254 has 12:A0:34:91:F2:4E

# ARP

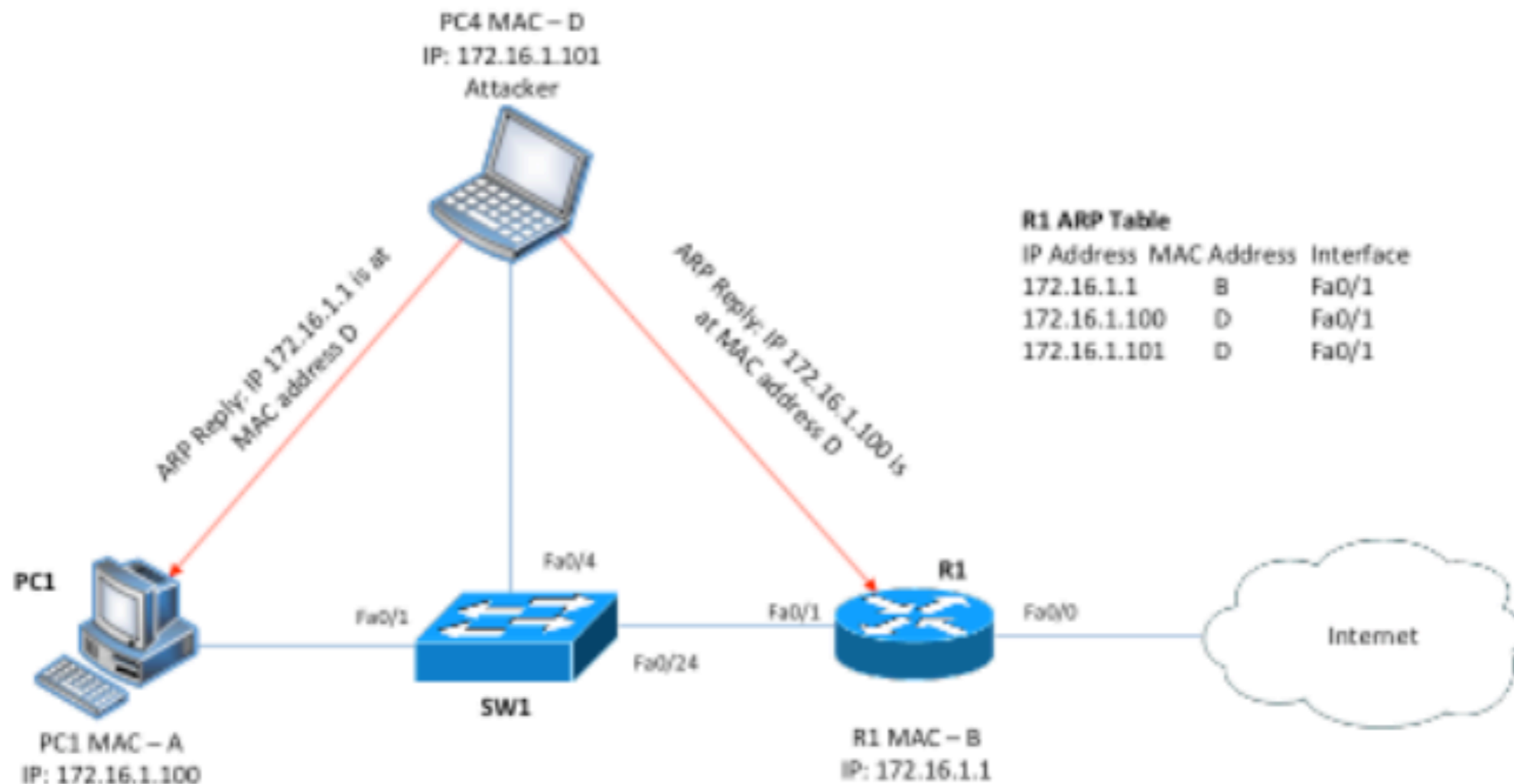
- ARP Cache
  - Static ARP cache:
    - Entries that are manually added
      - Can be used for ad-blocking
  - Dynamic ARP caches:
    - Hardware - IP address pairs added whenever an ARP packet is captured



```
thomasschwarz — -bash — 80x24
Last login: Mon Nov 13 15:39:30 on ttys001
MSCSs-MacBook-Pro-2:~ thomasschwarz$ arp -a
? (134.48.21.81) at a0:8c:fd:a9:d6:c3 on en3 ifscope [ethernet]
csdnet21.csd.mu.edu (134.48.21.100) at 0:2c:c8:7d:81:3f on en3 ifscope [ethernet]
]
pcp043633pcs.its.mu.edu (134.48.21.126) at a0:d3:c1:15:b:d4 on en3 ifscope [ethernet]
pcp043632pcs.its.mu.edu (134.48.21.127) at 58:ef:68:7b:b7:f8 on en3 ifscope [ethernet]
pcp043626pcs.its.mu.edu (134.48.21.133) at a0:8c:fd:a9:54:5b on en3 ifscope [ethernet]
? (134.48.21.179) at 88:51:fb:43:ab:b8 on en3 ifscope [ethernet]
? (134.48.21.187) at a0:8c:fd:a9:d6:18 on en3 ifscope [ethernet]
? (169.254.226.197) at bc:83:85:fa:33:7d on en3 [ethernet]
? (224.0.0.251) at 1:0:5e:0:0:fb on en3 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en3 ifscope permanent [ethernet]
MSCSs-MacBook-Pro-2:~ thomasschwarz$
```

# ARP-Poisoning

- Use unsolicited ARP messages to route messages to a gateway to an attacker and messages from a gateway to an attacker

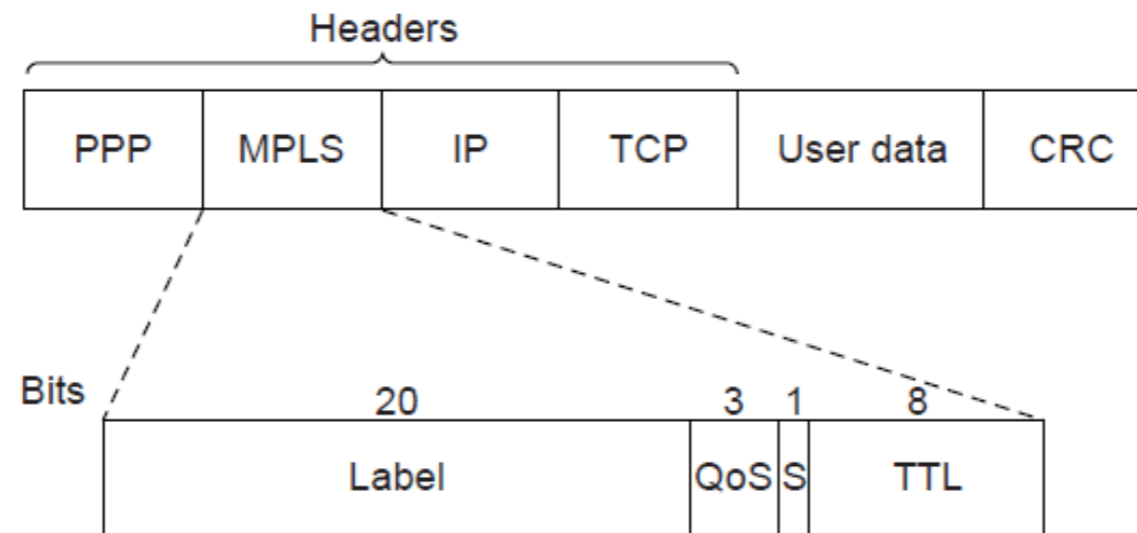


# Reverse Address Resolution Protocol (RARP)

- Reverse ARP
  - Get IP address based on MAC address
    - Needed if machine has no storage
      - (diskless workstations)
  - Client generates RARP request frame and broadcast it
  - RARP server sends RARP reply message with the desired information

# Label Switching and MPLPS

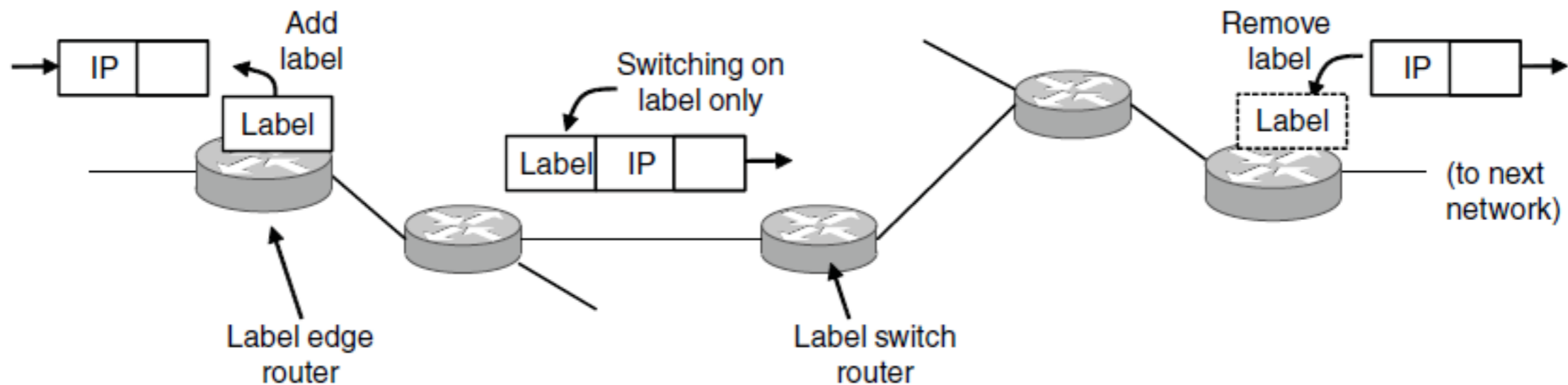
- MPLS (Multi-Protocol Label Switching) sends packets along established paths
  - Close to circuit switching: forwarding based on Label
  - ISPs can use for QoS
  - Path indicated with label below the IP layer



MPLS with PPP as framing protocol

# Label Switching and MPLS

- Label added based on IP address on entering an MPLS network (e.g., ISP) and removed when leaving it
  - Forwarding only uses label inside MPLS network



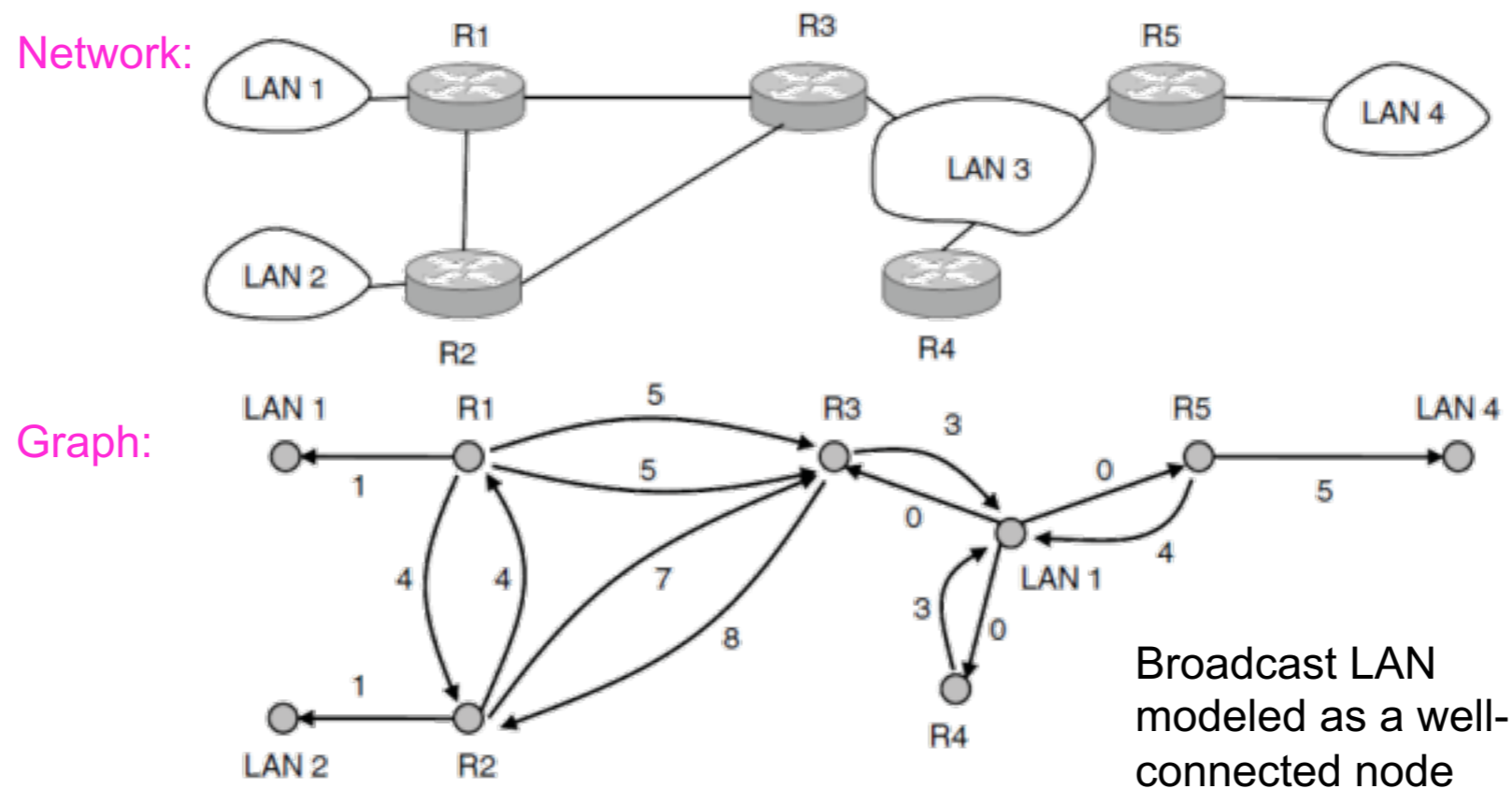


# MPLS Routing

- Only with Label Switched Routers (LSR)
- Label Edge Routers (LER) inspect IP packets and add labels
- All flows with the same label are in a Forwarding Equivalency Class (FEC)
- Routing & connection setup protocols are used for setting up routing

# Open Shortest Path First OSPF

- Interior routing protocol
  - OSPF computes routes for a single network (e.g., ISP)
  - Models network as a graph of weighted edges



# OSPF

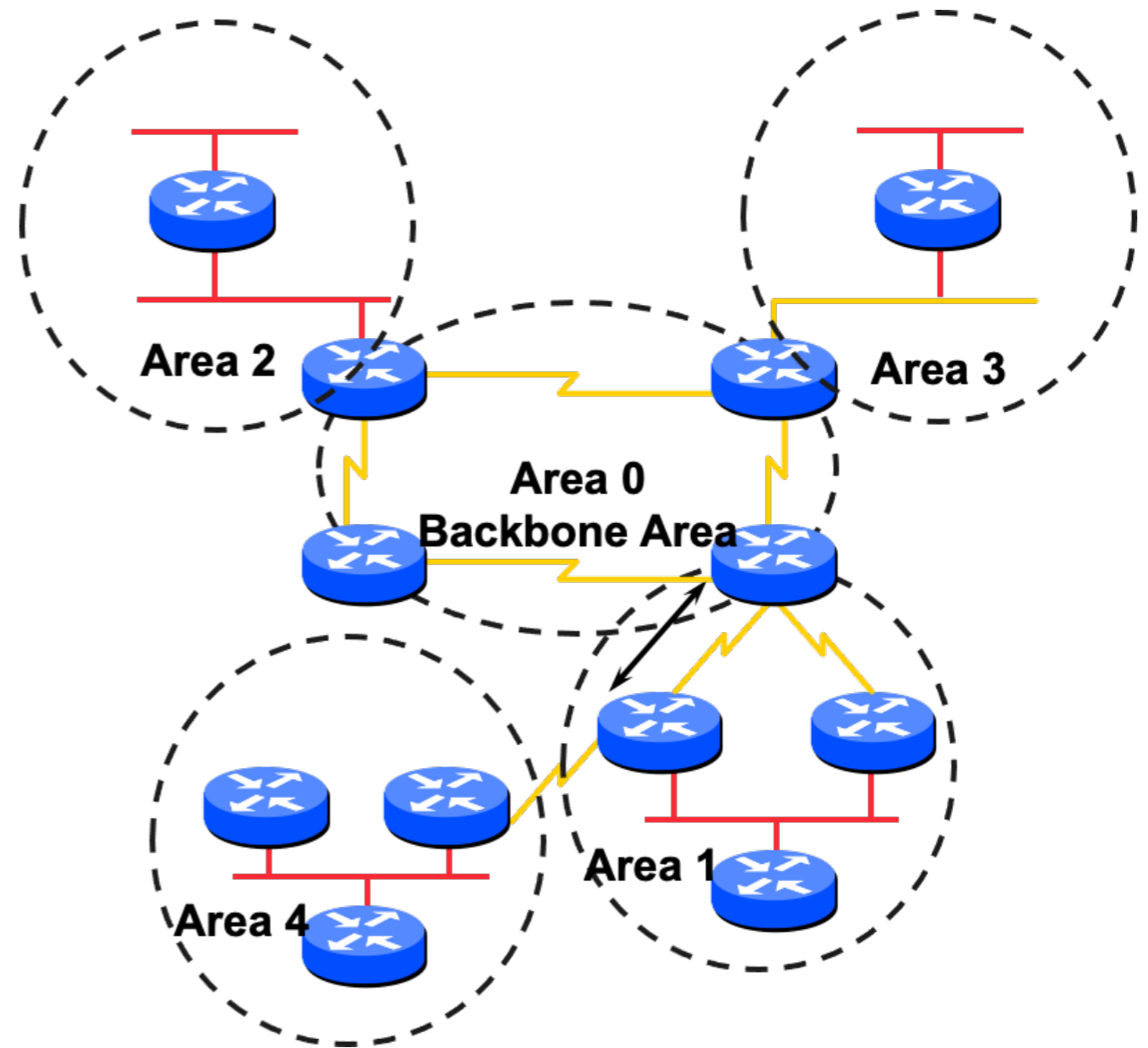
- Costs of links can be based on the type of service

# OSPF

- Autonomous System (AS)
  - Region administered by a single entity
- OSPF is for administering an AS
  - But AS can be too big
  - OSPF divides AS into **areas**

# OSPF

- One of the areas is the backbone area
- All other areas are connected to the backbone
- Can create “virtual links” to compensate for “awkward topologies”

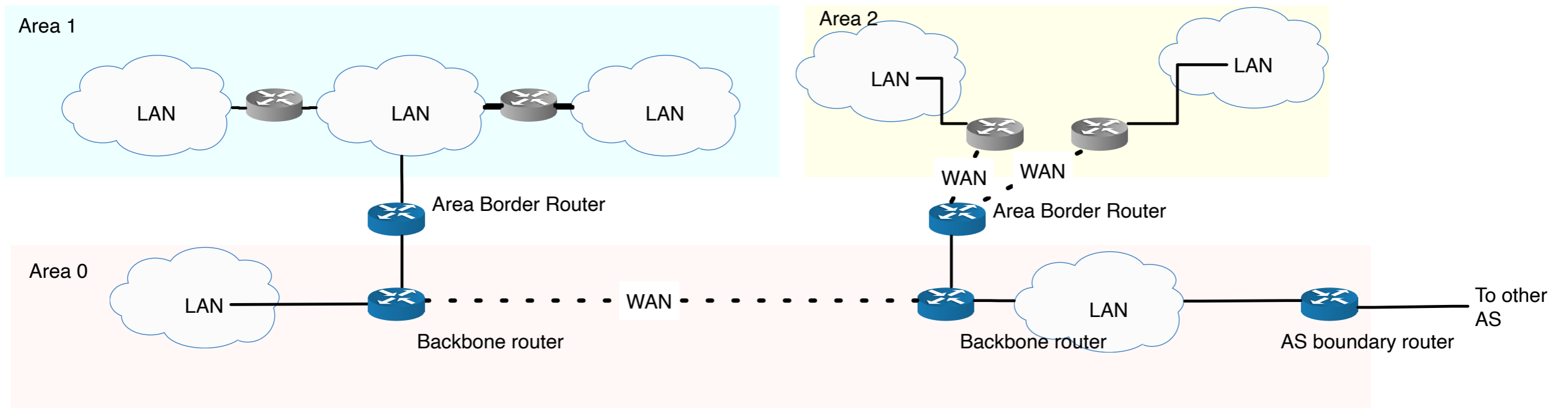


# OSPF

- Area 0 : backbone area with backbone routers
- Routers between areas are called area border routers
- Routers inside an area (other than Area 0) are called interior routers.
- Stub areas: only one path out of area

# OSPF

- Example



# OSPF

- Routers flood only their area with Link State Packets (LSP)
- There is **one** backbone area
  - Its job is to glue the other areas together
  - It passes the information collected by other areas to all other areas
- All routers calculate their forwarding tables based on LSPs



# OSPF

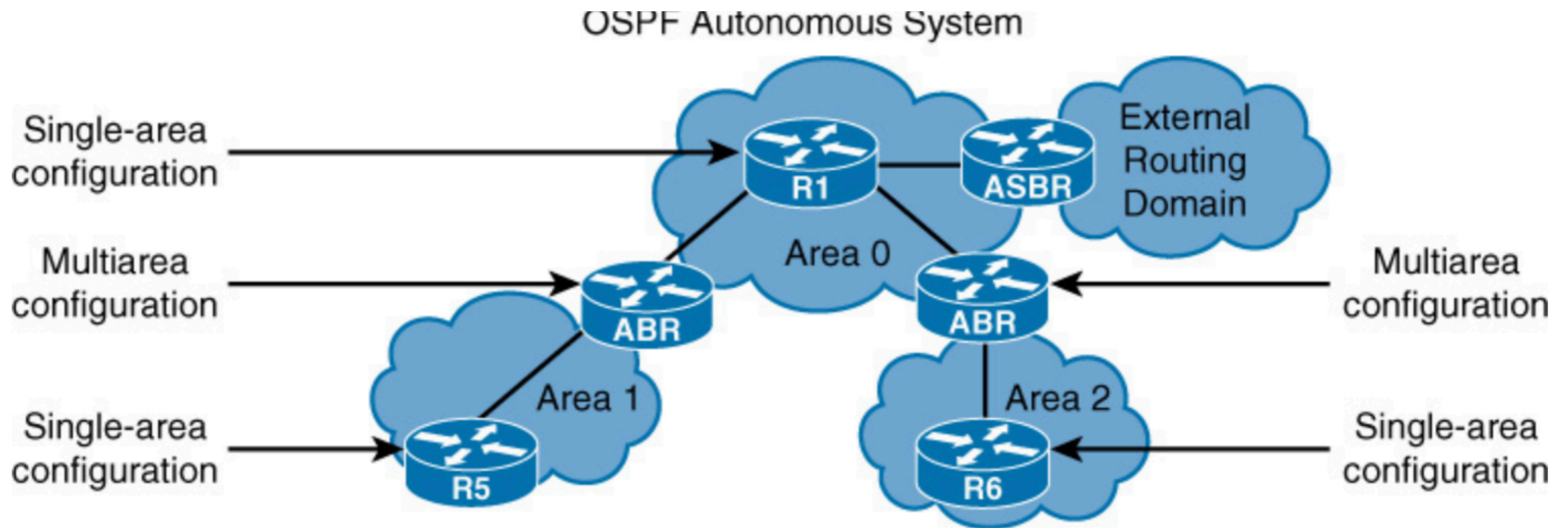
- Uses link-state routing
  - Uses messages to flood topology reliably
  - Runs Dijkstra's algorithm to compute routes

<b>Message type</b>	<b>Description</b>
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

# OSPF

- Links are more complicated than in graph theory
  - Router links: transient, stub, point-to-point links
  - Network links: advertises the network as a node
  - Summary link to network: Area border router advertises summary of links connected by the backbone
  - Summary link to AS: AS router advertises summary links from other AS to the backbone
  - External link: AS router advertises the existence of a single network outside the AS to the backbone area

# OSPF

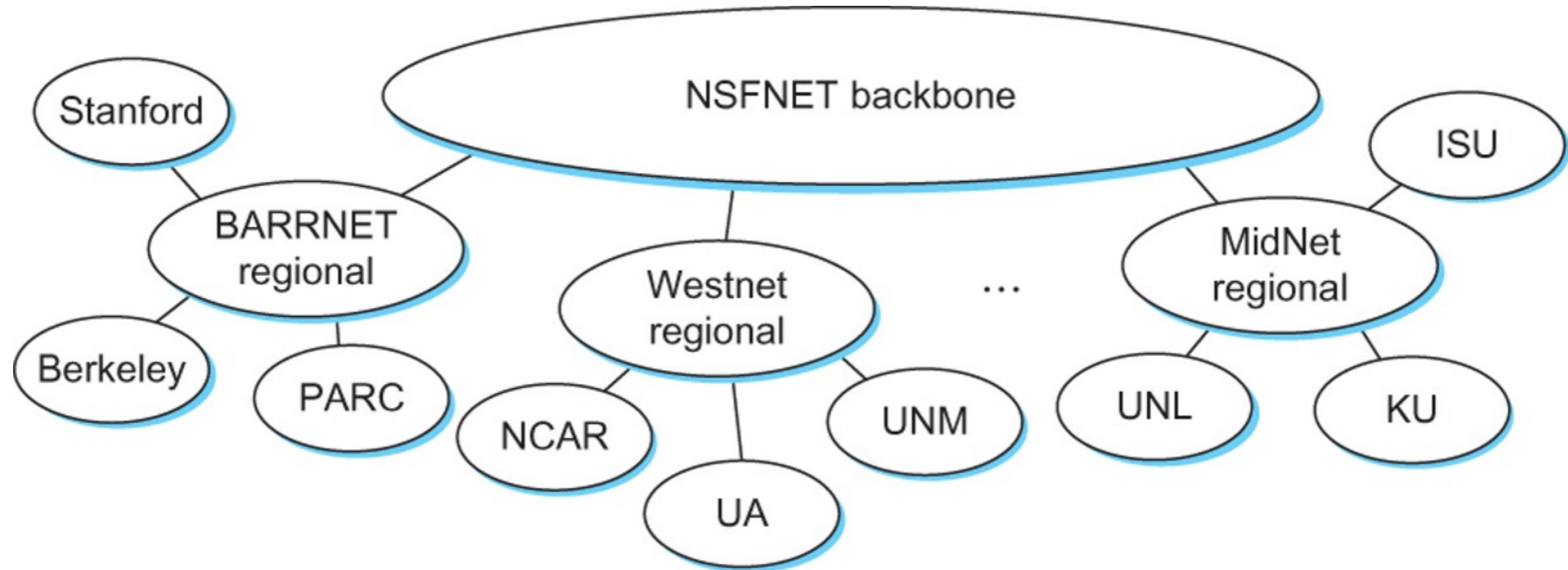


# Border Gateway Protocol

## BGP

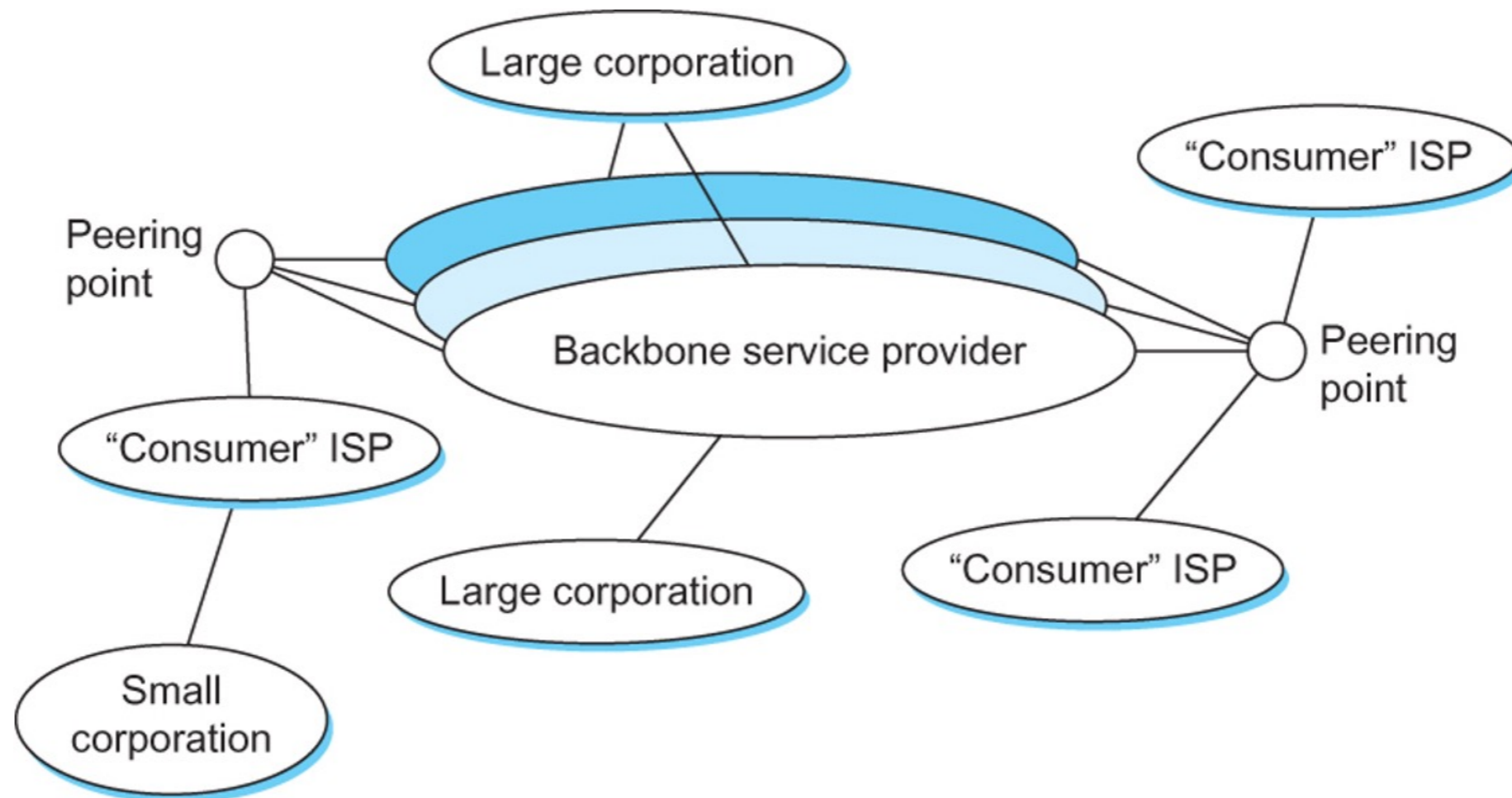
- BGP (Border Gateway Protocol) computes routes across interconnected, autonomous networks
  - Key role is to respect networks' policy constraints
  - Example policy constraints:
    - No commercial traffic for educational network
    - Never put Iraq on route starting at Pentagon
    - Choose cheaper network
    - Choose better performing network
    - Don't go from Apple to Google to Apple

# Global Internet



**Internet in 1990**

# The Global Internet



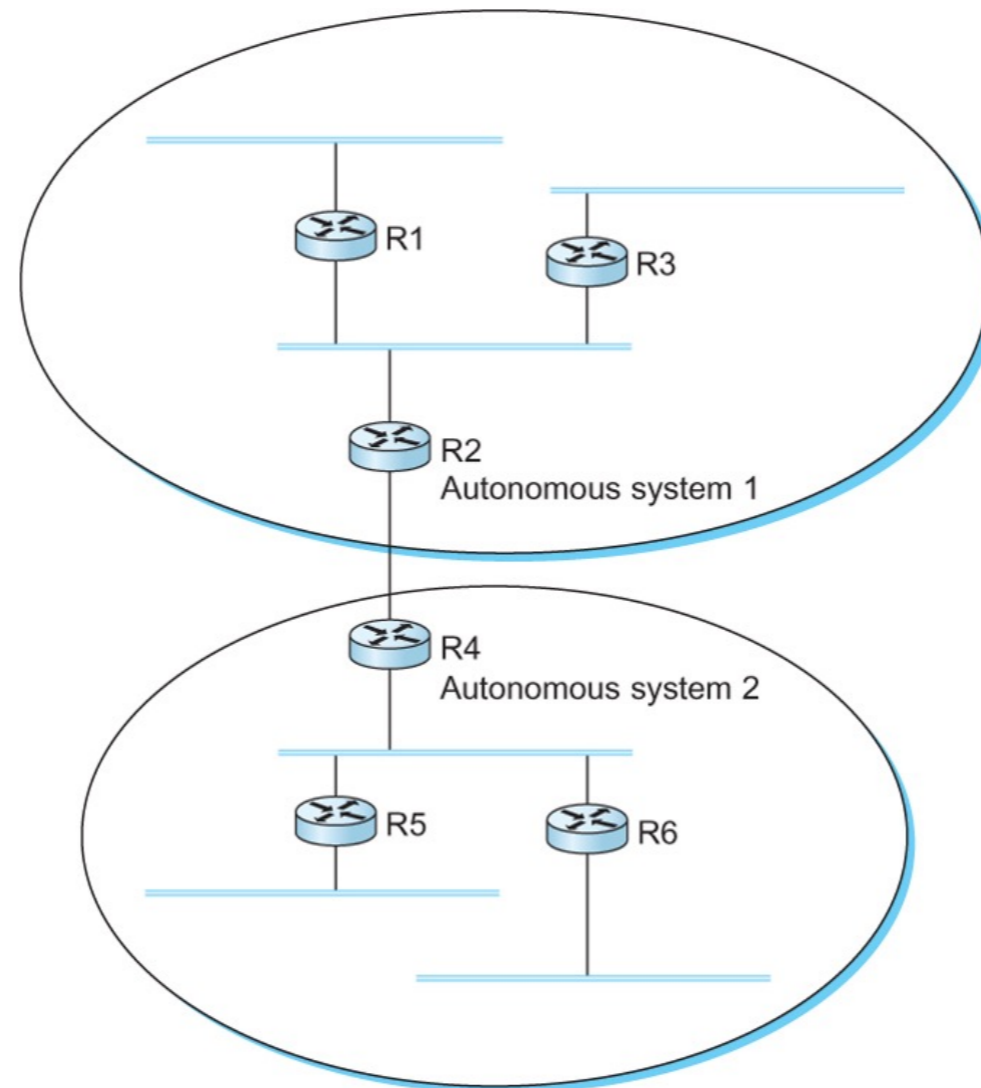
**A simple multiprovider internet**

# BGP: Border Gateway Protocol

- Internet consists of autonomous systems (AS)
  - Each AS is controlled by a single entity
  - Example AS:
    - University, company, backbone network
  - AS are also called “routing domain”
    - Has a set of IP addresses that it administers
- Each AS has a unique identifier
  - 16 bit number

# BGP

- Routing is done inside and between AS





# BGP

- Use hierarchical routing
  - Divide routing into two parts
    - Routing within a single AS: Intra-AS
      - Type of routing sole responsibility of AS
    - Routing between AS: Inter-AS
      - Routing uses internet-wide standards

# BGP

- Interior Gateway Protocol
  - Within an Autonomous System
  - Carries information about internal prefixes
  - Examples—OSPF, ISIS, EIGRP...
- Exterior Gateway Protocol
  - Used to convey routing information between ASes
  - De-coupled from the IGP
  - Current EGP is BGP4

# BGP

- History:
  - Exterior Gateway Protocol (EGP)
    - Forced a tree-like topology onto the internet
    - Good fit for ARPA-net
  - Border Gateway Protocol (BGP)
    - Allows arbitrary topology

# BGP

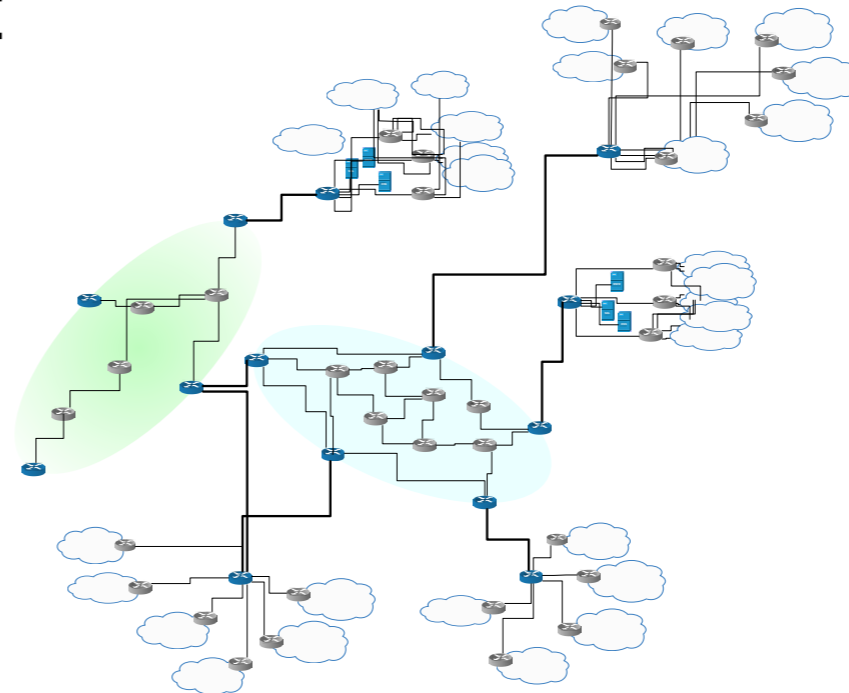
- Each AS routes traffic to and from its IP addresses
  - IP addresses are blocks of contiguous addresses
- Connected via dedicated links to other IPs
  - Negotiate reachability information using BGP
- BGP does not provide security guarantees
  - Vulnerable to attacks
  - Vulnerable to misconfigurations
    - Country-level censorship, cryptocurrency attacks, tracking users of anonymization networks

# BGP

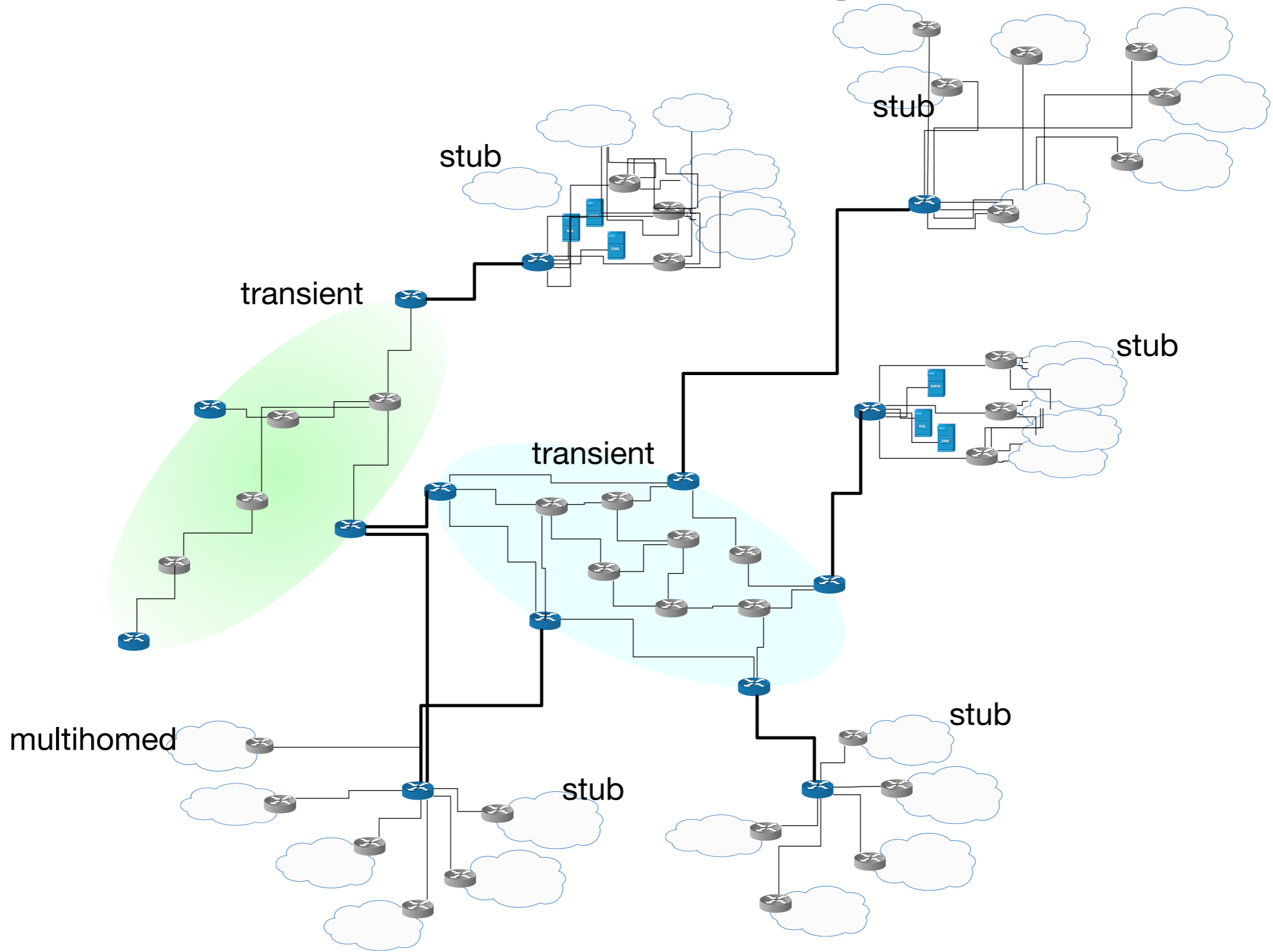
- AS interconnectivity based on confidential business agreements
  - Peer-to-peer agreements
    - Two AS of similar size agree to exchange traffic free of charge
  - Customer-provider agreement
    - AS pays another AS for connectivity

# BGP: AS Types

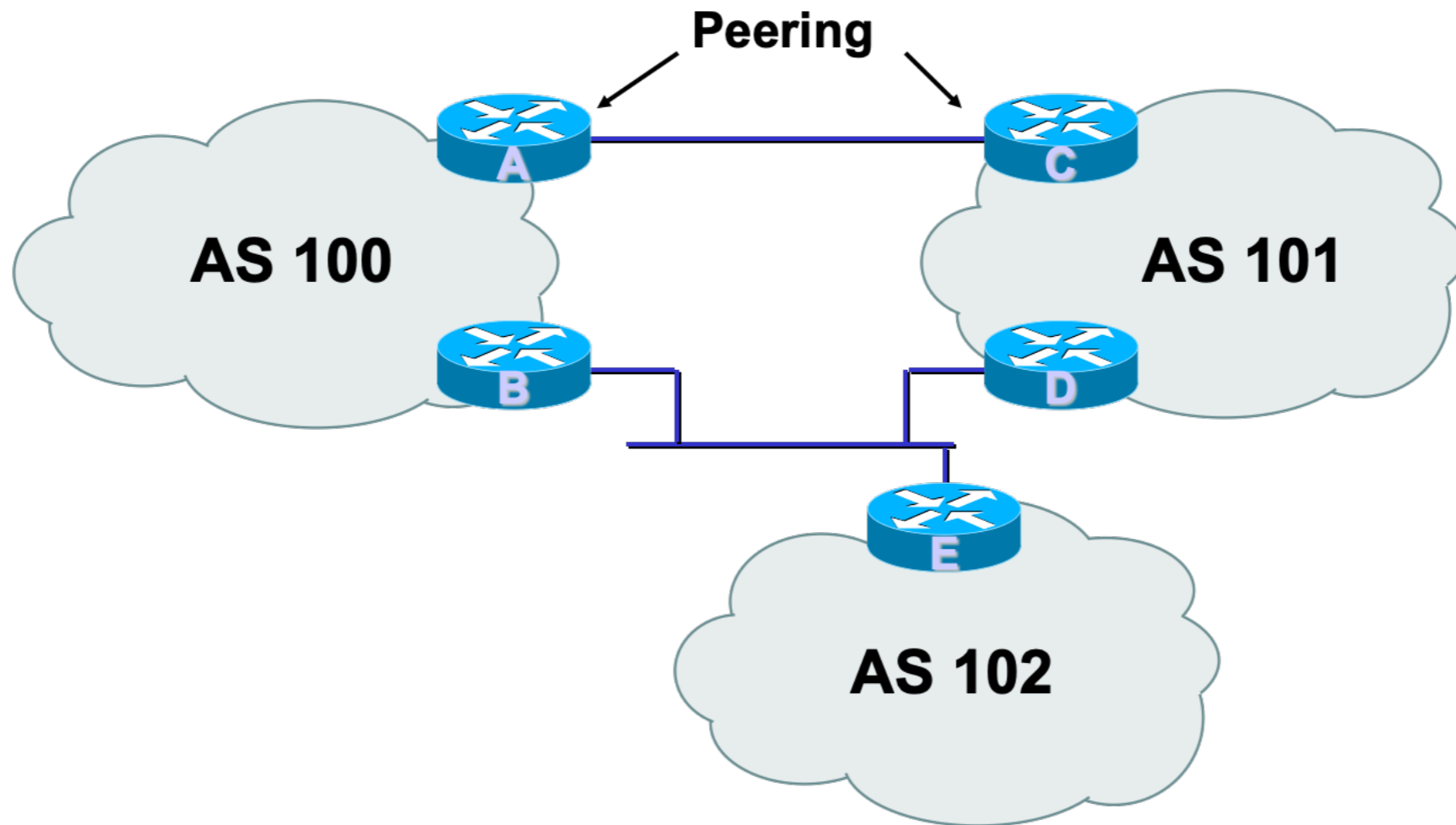
- Autonomous Systems (AS) can be:
  - Stub AS: Only one connection to another AS
  - Multihomed AS: More than one connection to other AS, but no traffic passes through it
  - Transient AS: Connects to more than one AS and has traffic run through it



# BGP: AS Types



# BGP



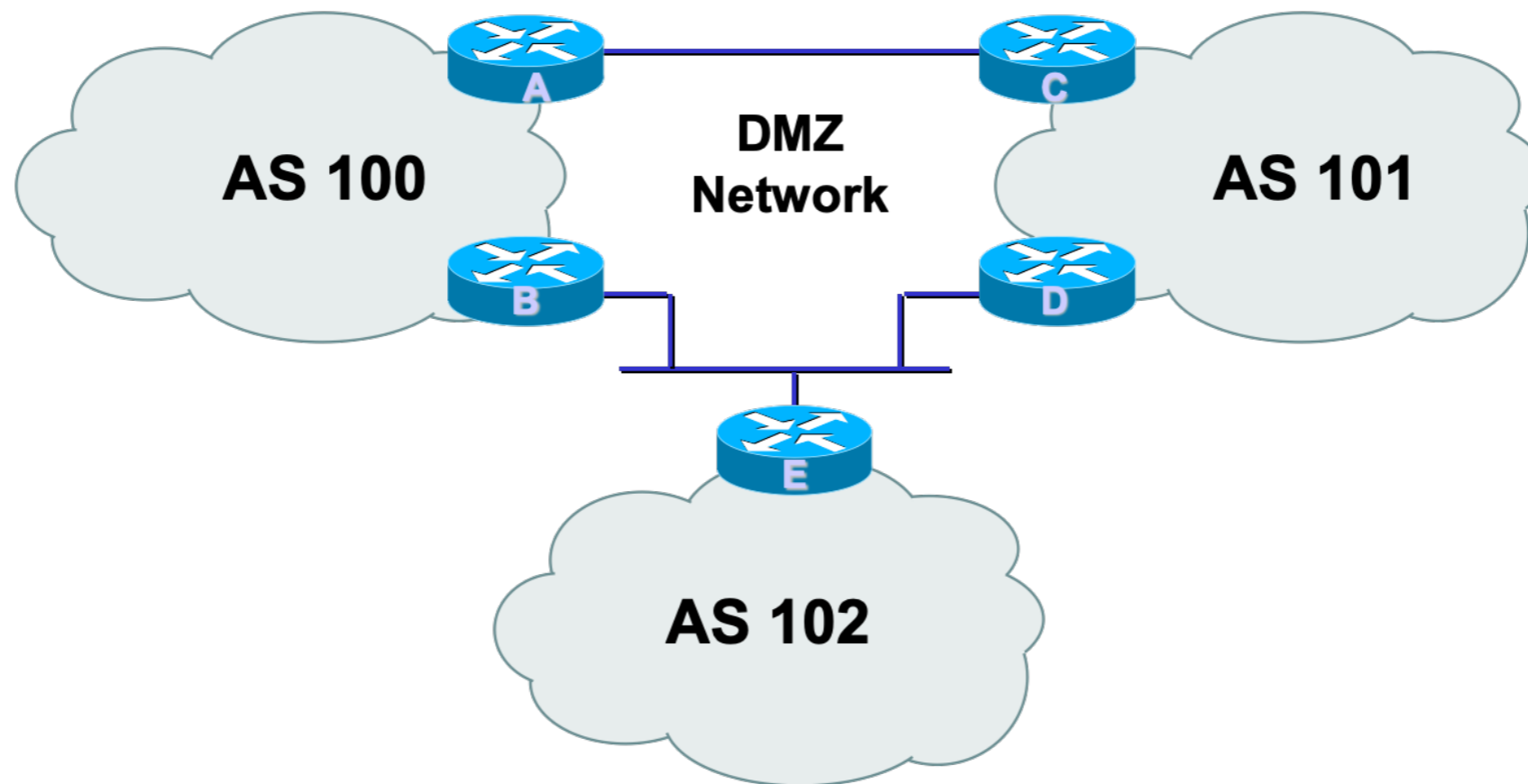


# BGP

- For networks in AS1 and AS2 to communicate:
  - AS1 must announce routes to AS2
  - AS2 must accept routes from AS1
  - AS2 must announce routes to AS1
  - AS1 must accept routes from AS2

# Demilitarized Zone

- Meaning 1: Area between firewall and the internet
- Meaning 2: Part of network between AS



# BGP

- BGP: Path-vector protocol
  - Construct paths by successively propagating advertisements between BGP peers
    - Advertisement includes list of AS on path
    - AS can favor paths:
      - *local preference, Multiple Exit Discriminator, AS prepending* (make path look longer)

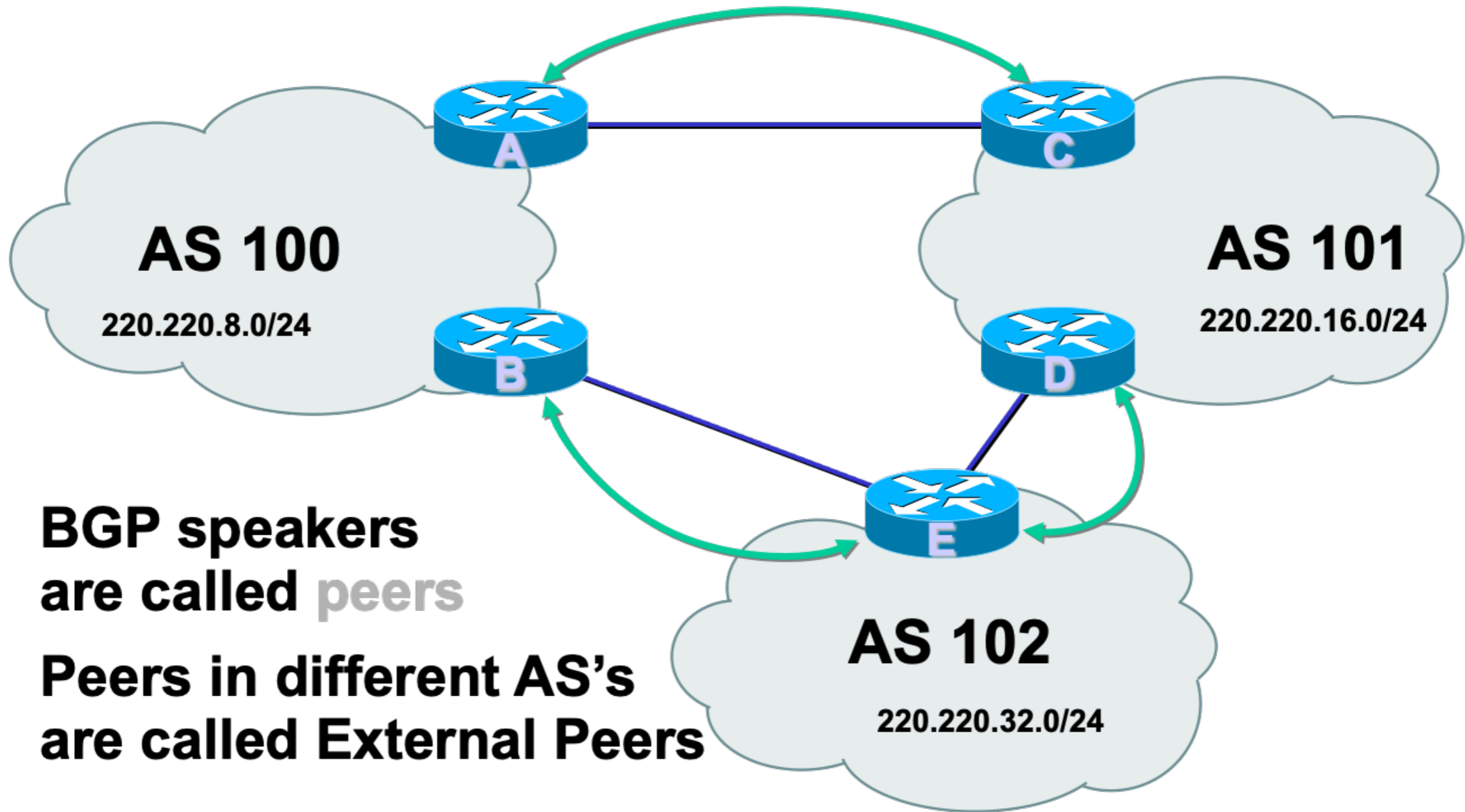
# BGP

- Processed with Network Layer Reachability Information
  - NLRI

# BGP

- Border routers
  - Use eBGP between different AS
    - Called BGP peers or BGP speakers
    - Connect via TCP on port 179
    - Connection should be direct
  - Use iBGP to connect to routers in the same AS
    - Connect via TCP on port 179
    - Create an overlay network that connects all BGP peers in the same AS

# eBGP



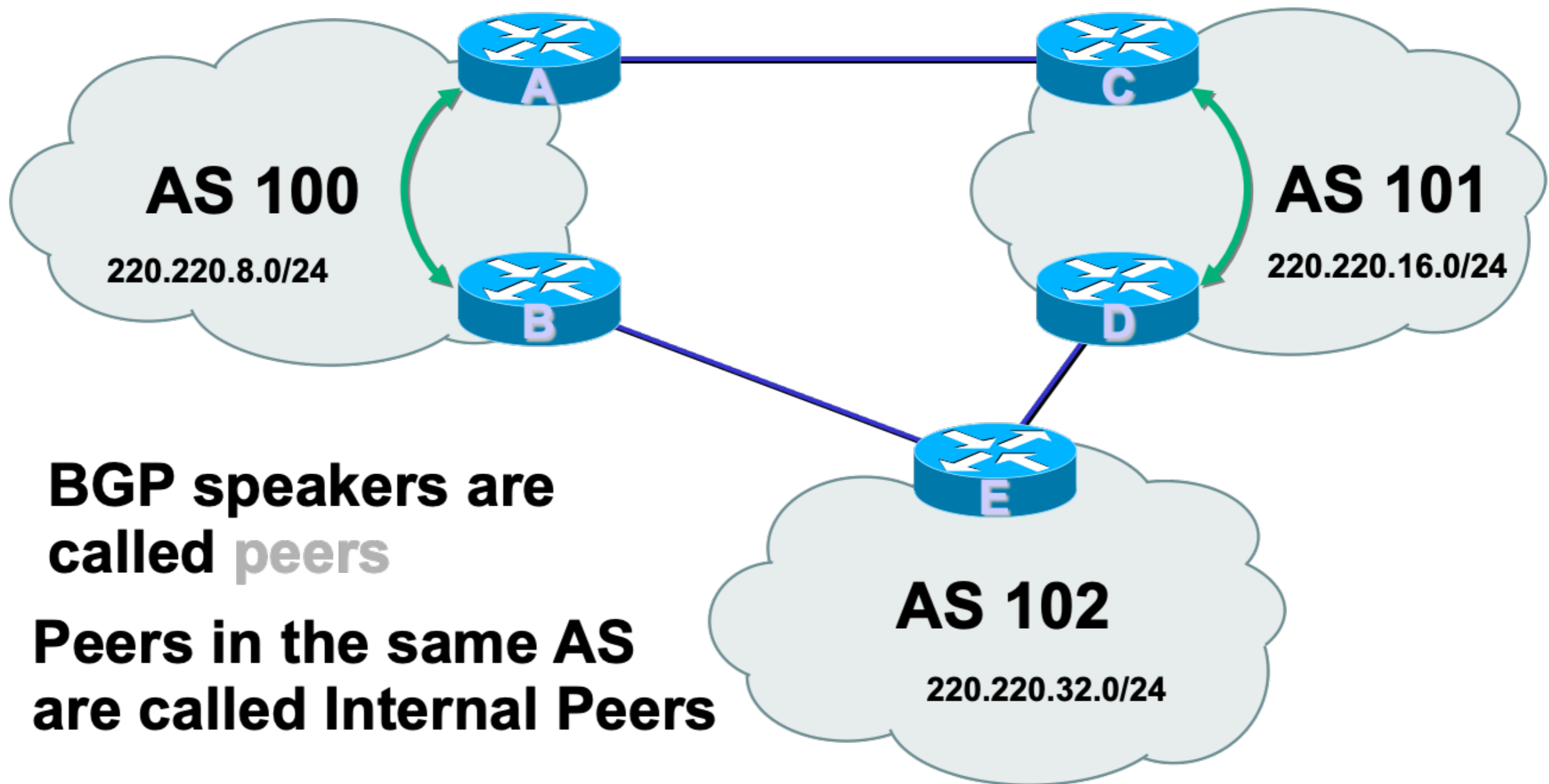
**BGP speakers**  
are called **peers**

**Peers in different AS's**  
are called **External Peers**



Note: eBGP Peers normally should be directly connected.

# iBGP



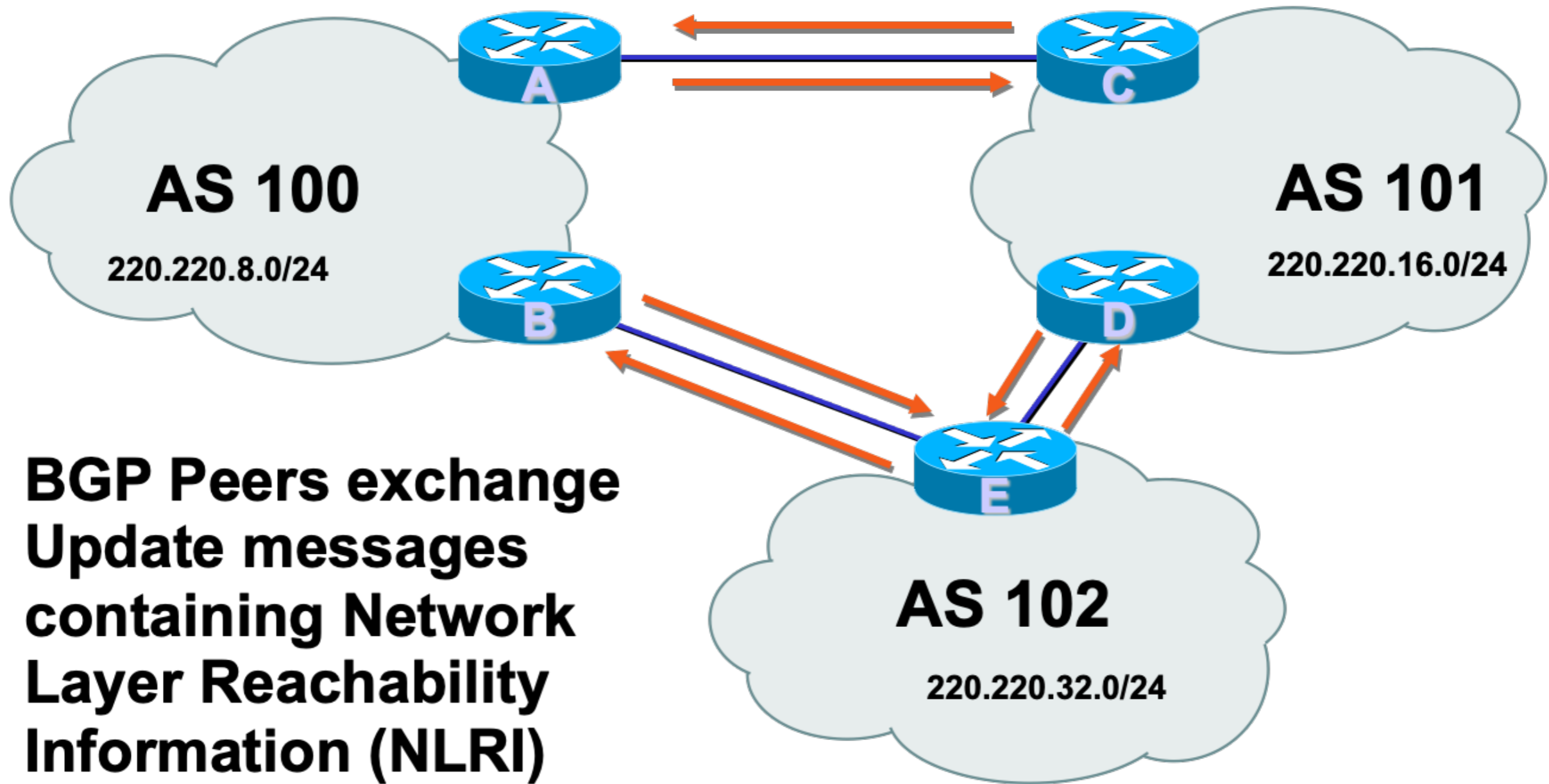
**BGP speakers are called peers**

**Peers in the same AS are called Internal Peers**



**Note: iBGP Peers don't have to be directly connected.**

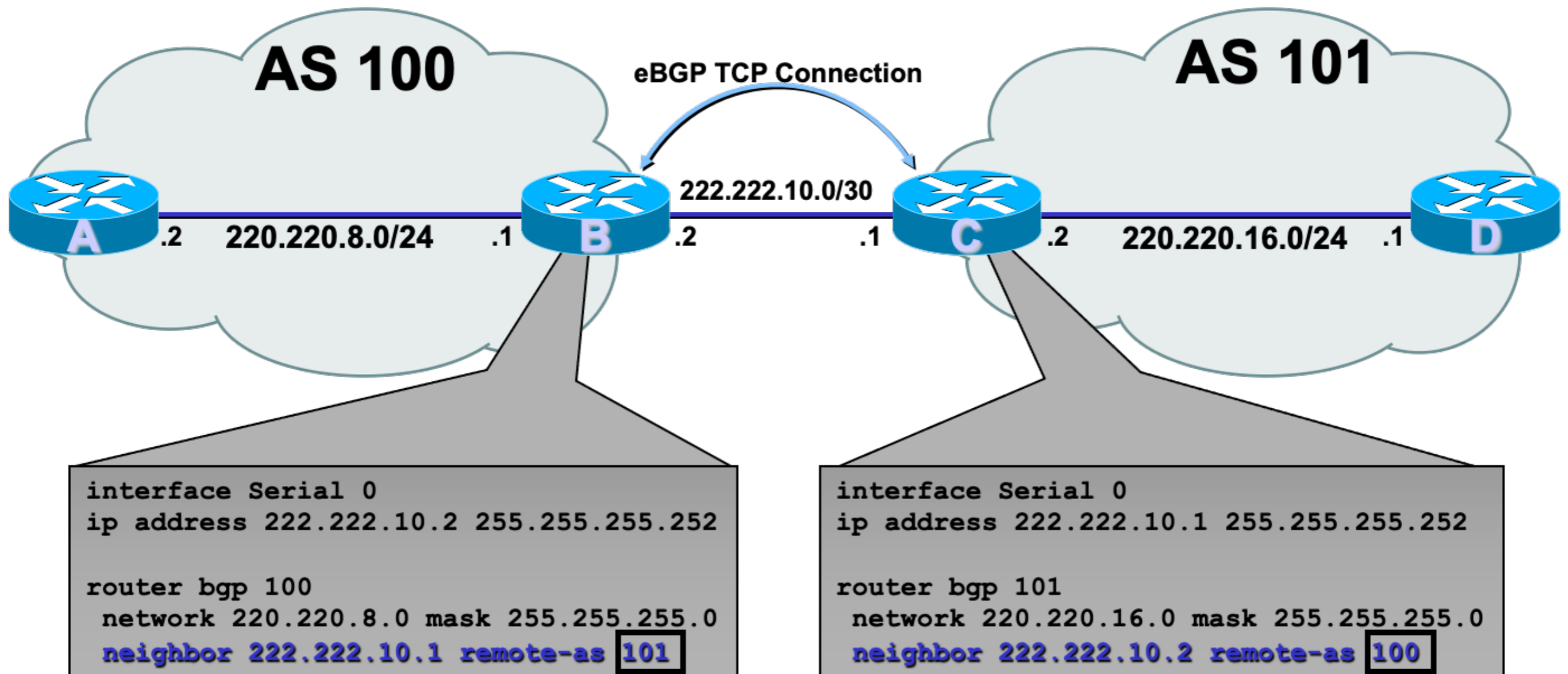
# BGP



BGP Update  
Messages →

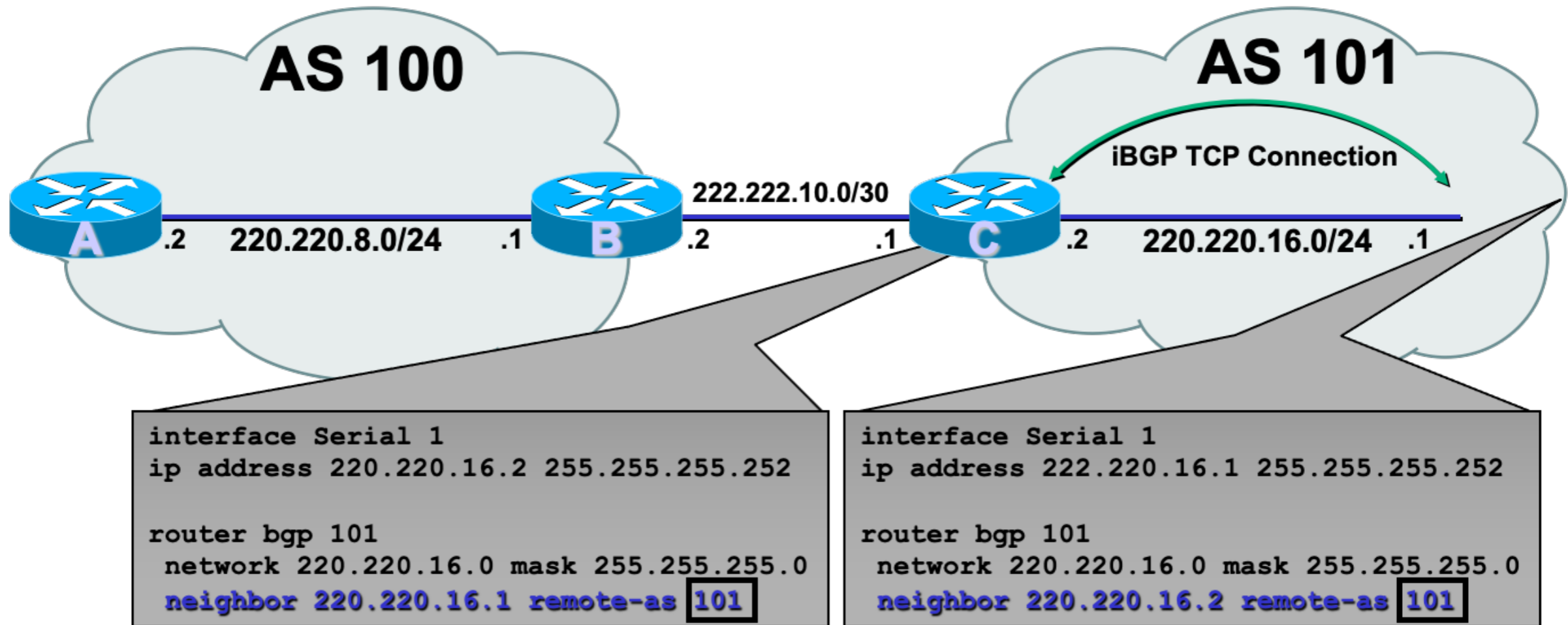


# BGP



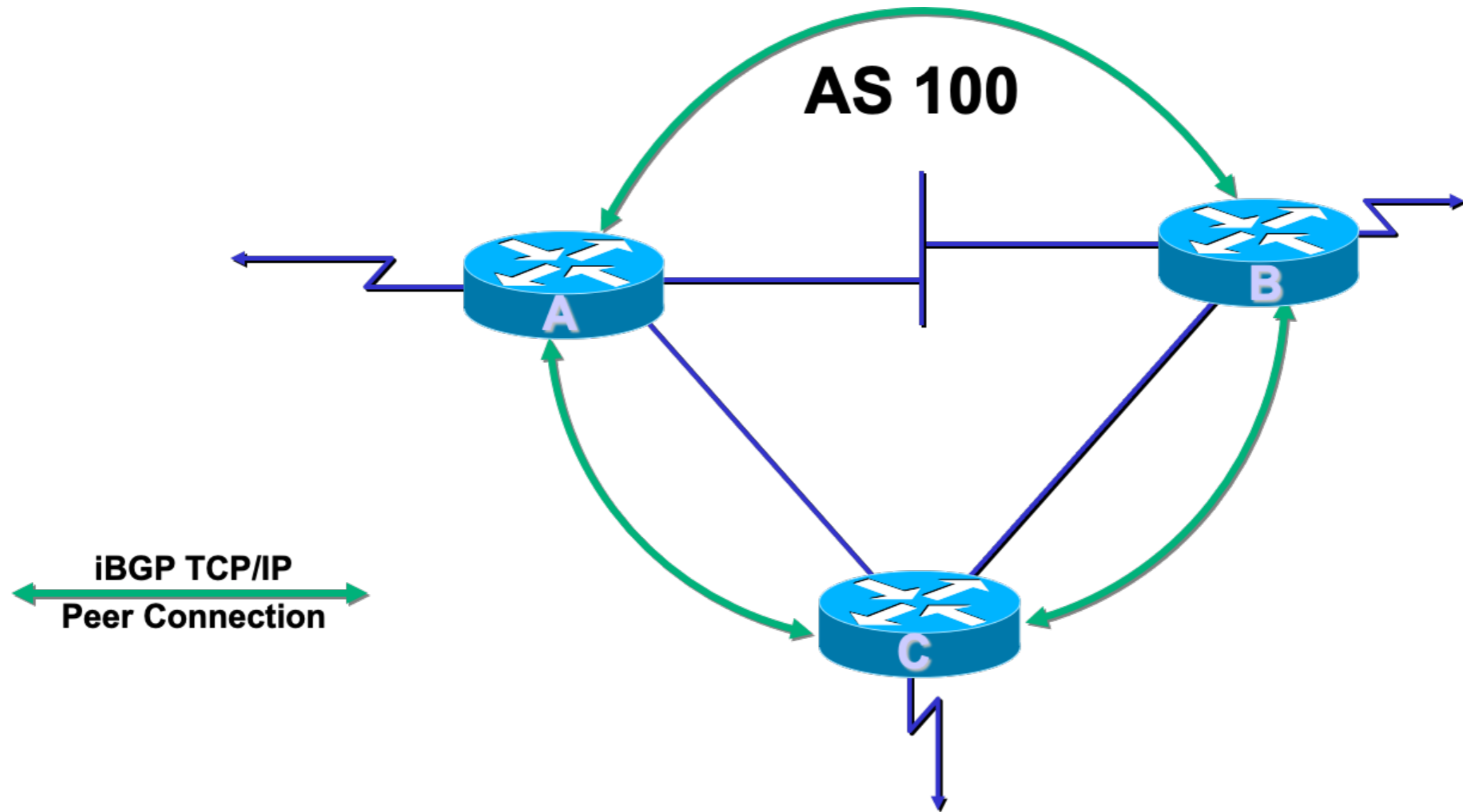
- **BGP Peering sessions are established using the BGP “neighbor” configuration command**
  - External (eBGP) is configured when AS numbers are different

# BGP



- **BGP Peering sessions are established using the BGP “neighbor” configuration command**
  - External (eBGP) is configured when AS numbers are different
  - Internal (iBGP) is configured when AS numbers are same

# BGP



- Each iBGP speaker must peer with every other iBGP speaker in the AS

# BGP

- BGP peers combine information from eBGP and iBGP in a single routing table

# BGP

- Processing a BGP advertisement:
  - Import rules
    - which route to consider
  - Path selection
    - which route to select
  - Export rules (“Valley free rules”)
    - which routes to advertise

# BGP

- Egress traffic depends on:
  - Route availability
  - Route acceptance
  - Policy and tuning
  - Peering and transit agreements
- Ingress traffic depends on
  - What information you send and to who
  - Based on your addressing and ASes
  - Based on others' policy

# BGP

- Types of Routes
  - Static Routes
    - configured manually
  - Connected Routes
    - created automatically when an interface is 'up'
  - Interior Routes
    - Routes within an AS
  - Exterior Routes
    - Routes exterior to AS

# BGP

- Autonomous System Number
  - Globally unique identifiers for IP networks
  - 2-byte only AS number range: 0 — 65535
  - 4-byte only AS number range: 65,536 — 4,294,967,295

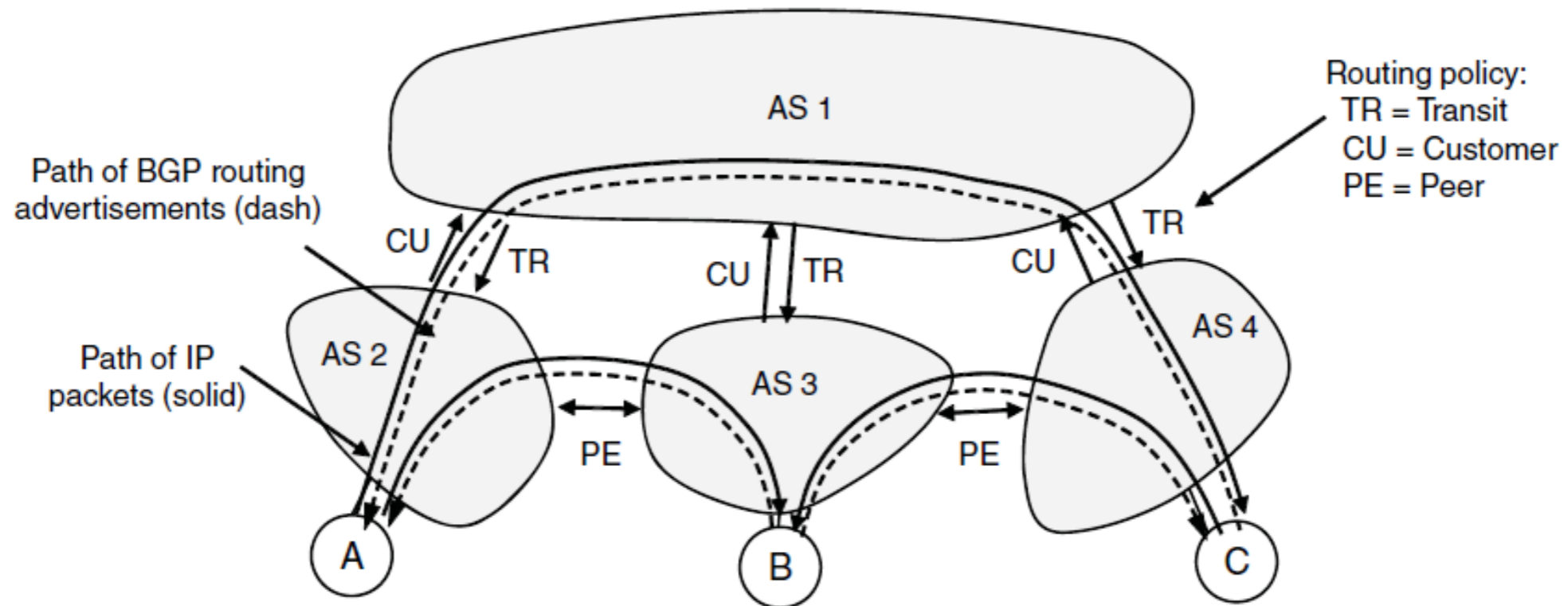


# BGP

- Path vector routing protocol:
  - Route is a collection of AS numbers
    - E.g.: {65001 65002 65003 65007}
    -

# BGP

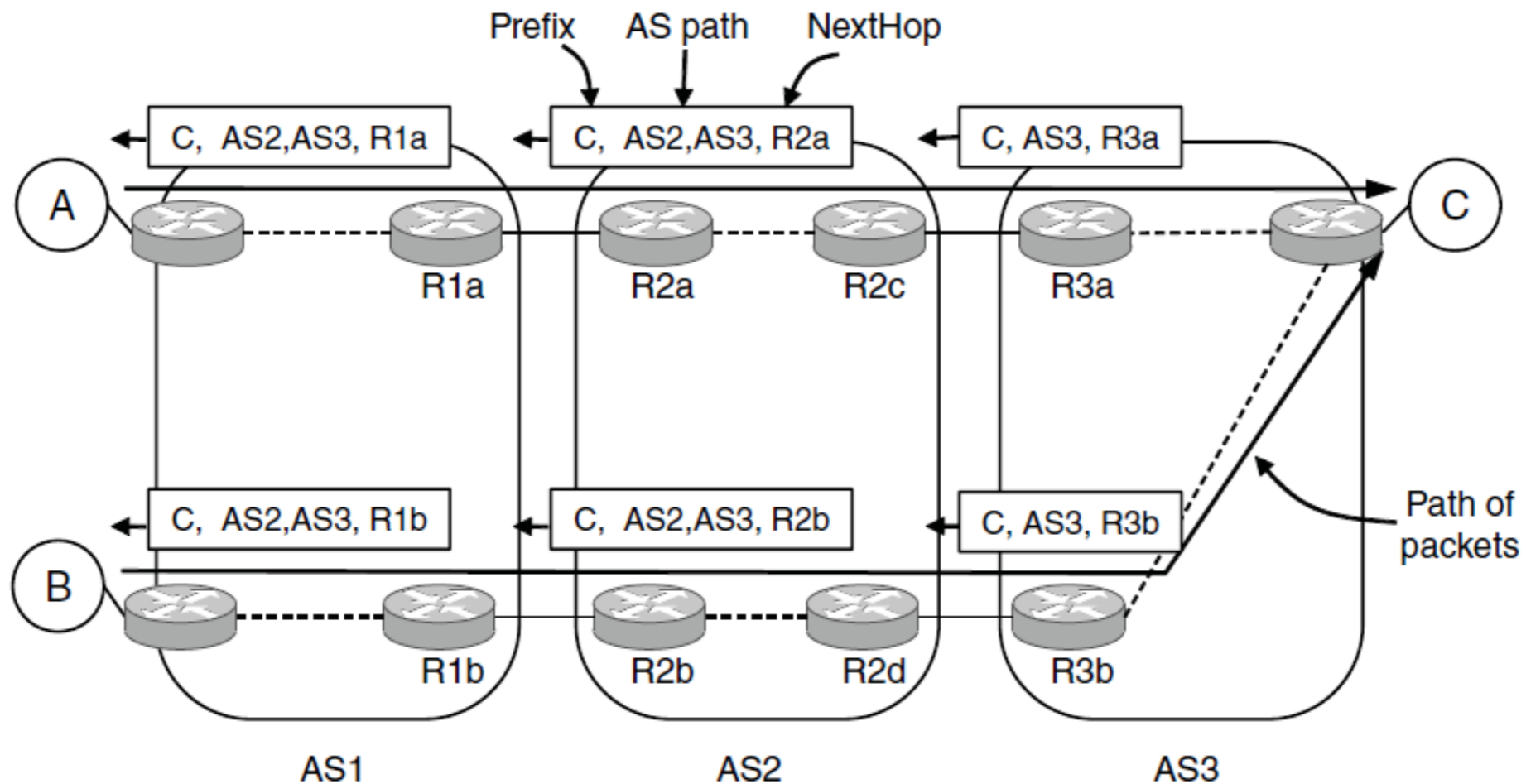
- Common policy distinction is transit vs. peering:
  - Transit carries traffic for pay; peers for mutual benefit
  - AS1 carries AS2↔AS4 (Transit) but not AS3 (Peer)



- Use Internet eXchange Points to connect with other ISP

# BGP

- BGP propagates messages along policy-compliant routes
- Message has prefix, AS path (to detect loops) and next-hop IP (to send over the local network)



# Internet Multicasting

- Groups have a reserved IP address range (class D)
  - Membership in a group handled by IGMP (Internet Group Management Protocol) that runs at routers
- Routes computed by protocols such as PIM:
  - Dense mode uses RPF with pruning
  - Sparse mode uses core-based trees
- IP multicasting is not widely used except within a single network, e.g., datacenter, cable TV network.

# Mobile IP

- Mobile hosts can be reached at fixed IP via a home agent
  - Home agent tunnels packets to reach the mobile host; reply can optimize path for subsequent packets
  - No changes to routers or fixed hosts

