Network Layer
Chapter 5

- ▶ Design Issues
- ▶ Routing Algorithms
- ▶ Congestion Control
- ▶ Quality of Service
- ▶ Internetworking
- ▶ Network Layer of the Internet

Revised: August 2011

The Network Layer

Responsible for delivering packets between endpoints over multiple links

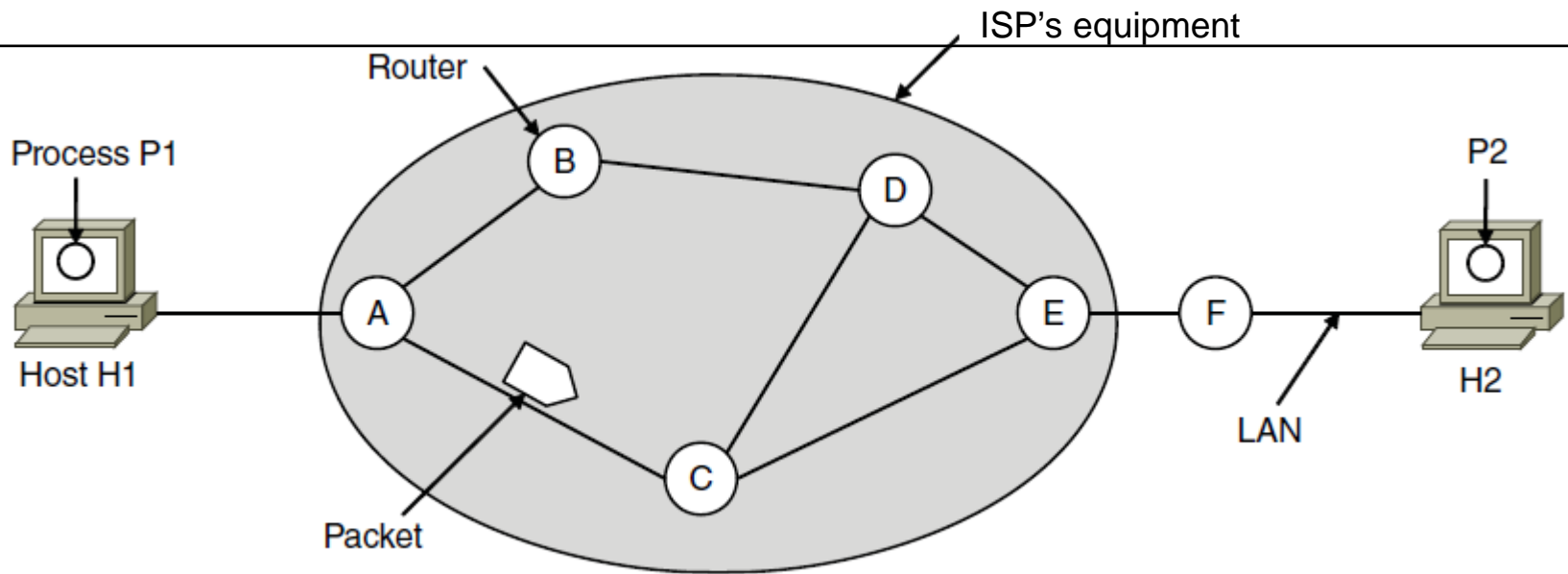| Application |
| --- |
| Transport |
| Network |
| Link |
| Physical |

Design Issues

- Store-and-forward packet switching »
- Connectionless service – datagrams »
- Connection-oriented service – virtual circuits »
- Comparison of virtual-circuits and datagrams »

Store-and-Forward Packet Switching

Hosts send packets into the network; packets are forwarded by routers

ISP's equipment

CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011

Connectionless Service – Datagrams

▶ Packet is forwarded using destination address inside it

▶ Different packets may take different paths
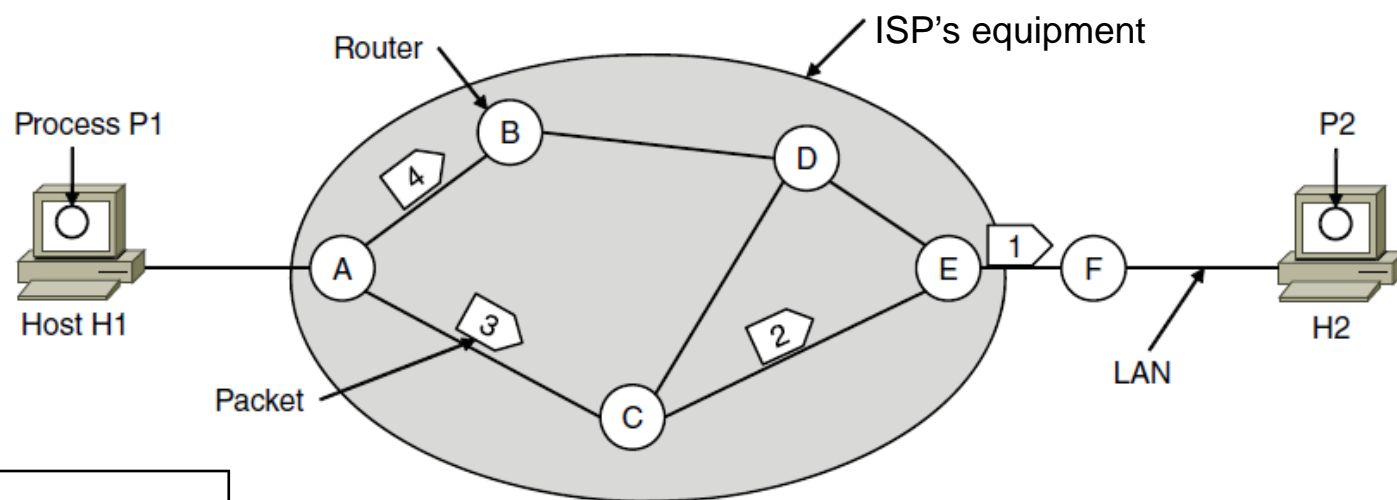
A's table (initially)

A	
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	
B	B
C	C
D	B
E	D
F	D

C's Table

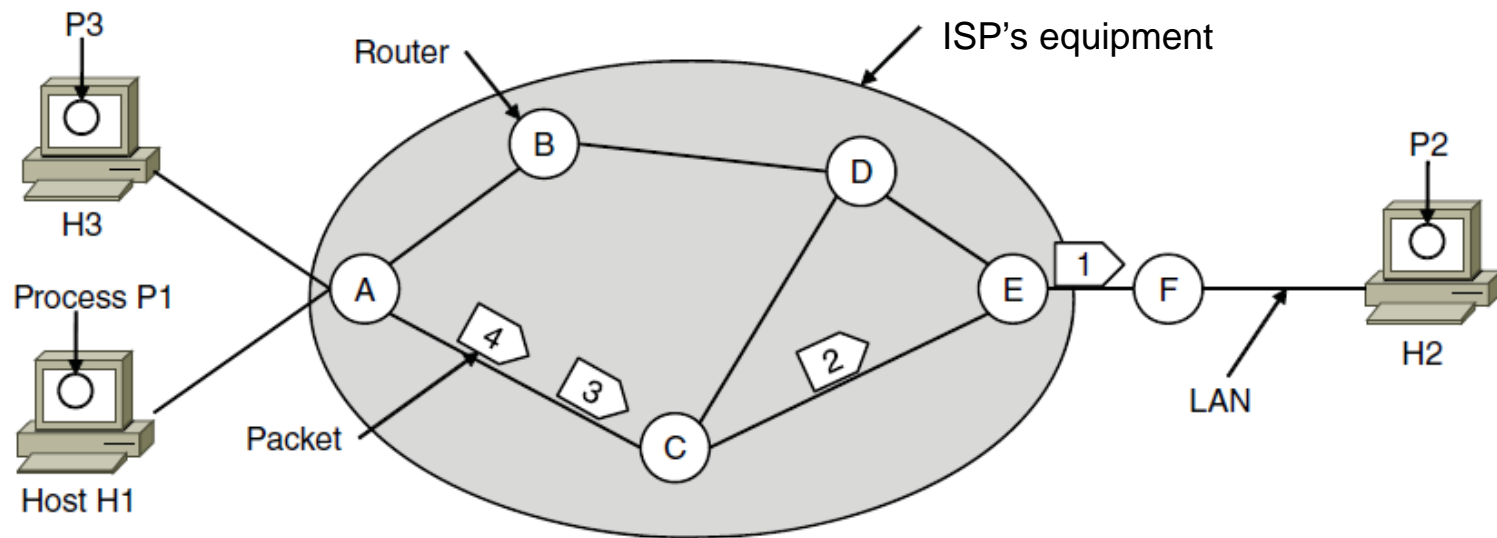| A	A
B	A
C	
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	
F	F

Dest. Line

Connection-Oriented – Virtual Circuits

▶ Packet is forwarded along a virtual circuit using tag inside it

   ▶ Virtual circuit (VC) is set up ahead of time

Comparison of Virtual-Circuits & Datagrams

| 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

Routing Algorithms (1)

- ▶ Optimality principle »
- ▶ Shortest path algorithm »
- ▶ Flooding »
- ▶ Distance vector routing »
- ▶ Link state routing »
- ▶ Hierarchical routing »
- ▶ Broadcast routing »
- ▶ Multicast routing »
- ▶ Anycast routing »
- ▶ Routing for mobile hosts »
- ▶ Routing in ad hoc networks »

Routing Algorithms (2)

- **<u>Routing</u>** 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)
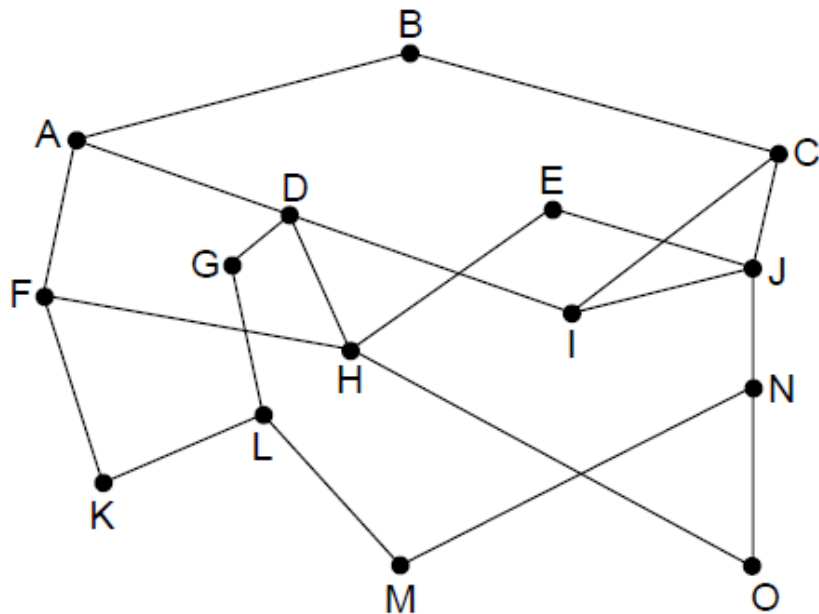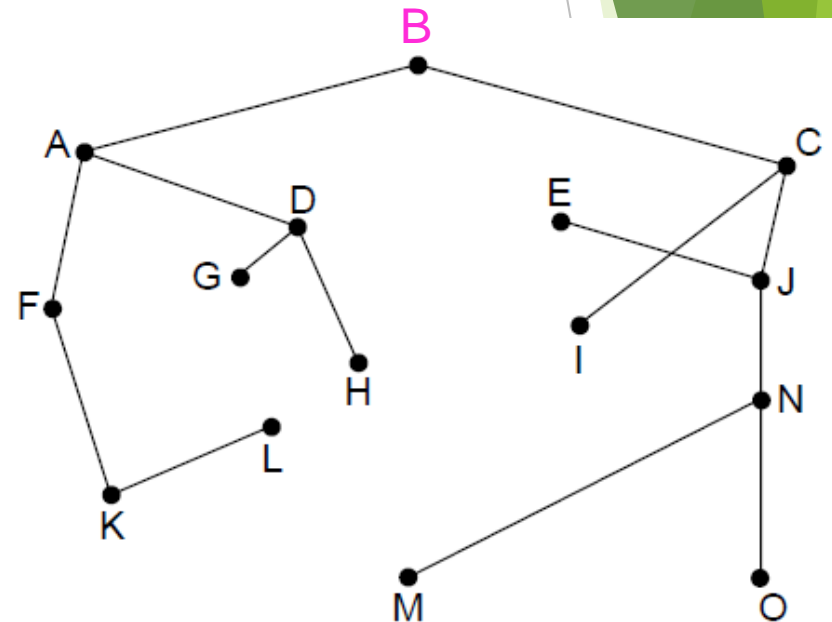 - Update routes for changes in topology (e.g., failures)
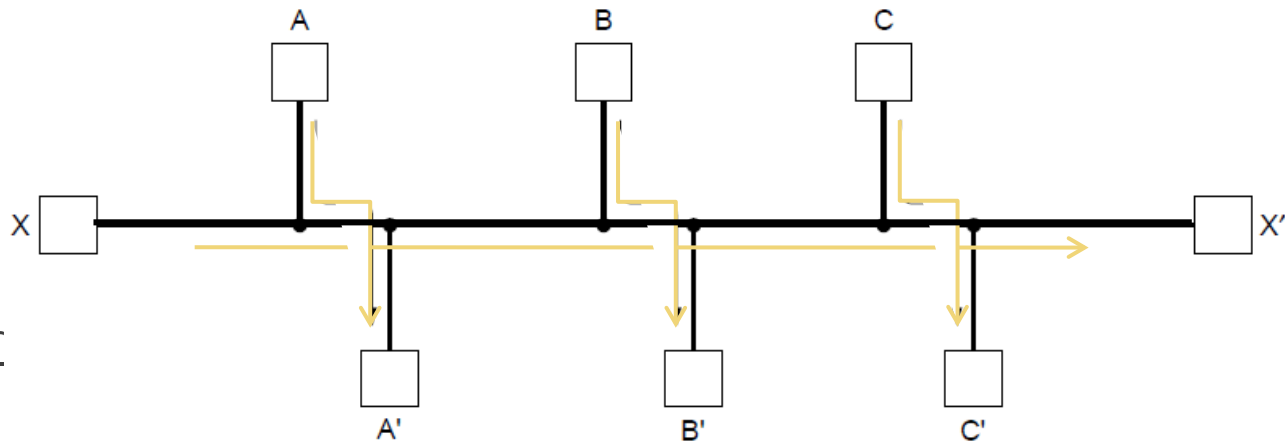
- <u>For</u>

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 <u>sink tree</u>

▶ Best means fewest hops in the example

Network

Sink tree of best paths to router B
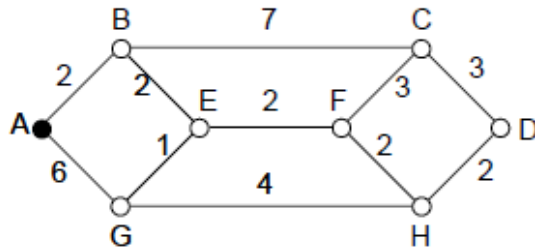
Shortest Path Algorithm (1)

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
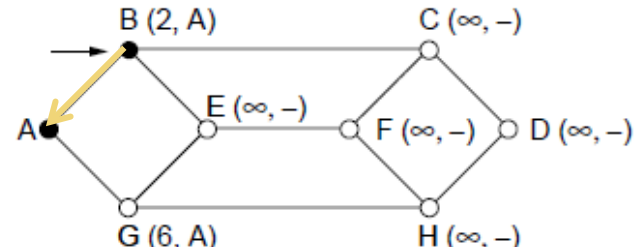- ▶ Using weights of 1 gives paths with fewest hops

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
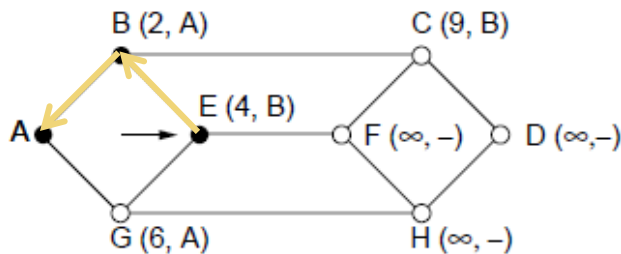- ▶ Repeat until all nodes are in the sink tree
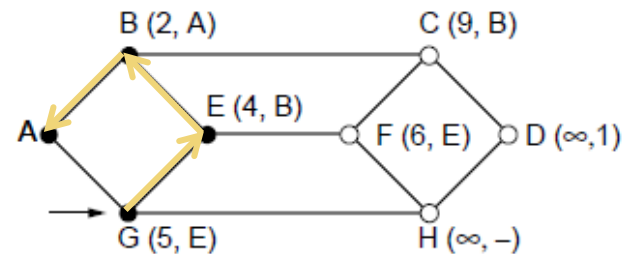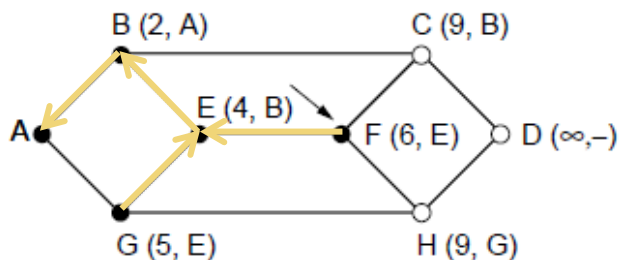
Shortest Path Algorithm (2)

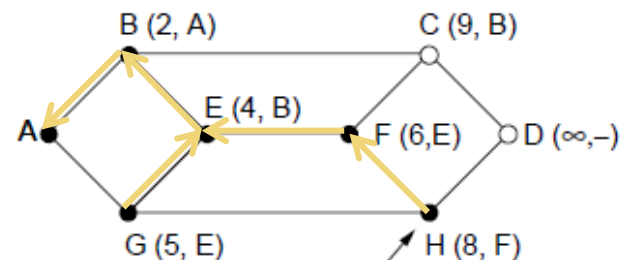A network and first five steps in computing the shortest paths from A to D. Pink arrows show the sink tree so far.

Shortest Path Algorithm (3)

. . .

```
for (p = &state[0]; p < &state[n]; p++) {
    p->predecessor = −1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0;  state[t].label = permanent;
k = t;
do {
    for (i = 0; i < n; i++)
        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];
            }
        }
```

. . .

Start with the sink, all other nodes are unreachable

Relaxation step. Lower distance to nodes linked to newest member of the sink tree

Shortest Path Algorithm (4)

. . .

```
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);
```

Find the lowest distance, add it to the sink tree, and repeat until done

Flooding

A simple 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
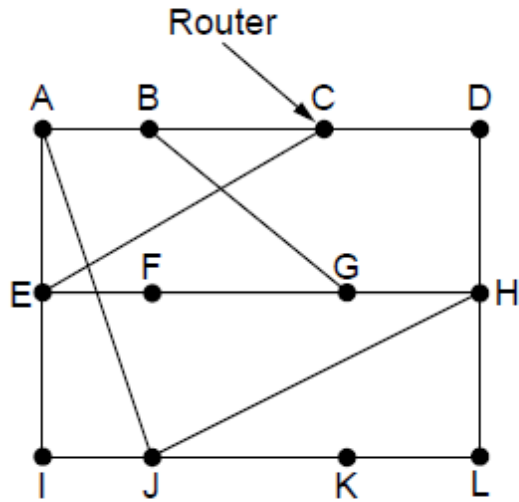
Distance Vector Routing (1)

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 (2)

Network

Vectors received at J from Neighbors A, I, H and K

The Count-to-Infinity Problem

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

A	B	C	D	E	
●	●	●	●	●	Initially
	1	●	●	●	After 1 exchange
	1	2	●	●	After 2 exchanges
	1	2	3	●	After 3 exchanges
	1	2	3	4	After 4 exchanges

Good news of a path to *A* spreads quickly

A	B	C	D	E	
● X	●	●	●	●	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
	⋮	⋮	⋮		
	●	●	●	●	

Bad news of no path to *A* is learned slowly

Link State Routing (1)

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 (2) – LSPs

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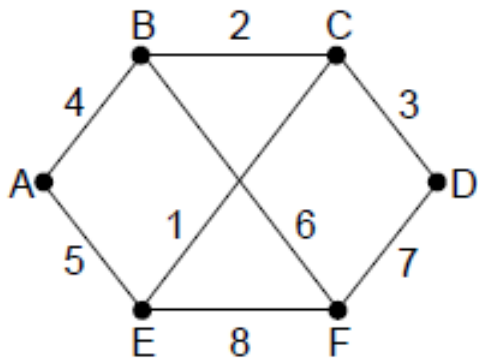
Network   LSP for each node

Link State Routing (3) – Reliable Flooding

Seq. number and age are used for reliable flooding

▶ New LSPs are acknowledged on the lines they are received and sent on all other lines

▶ Example shows the LSP database at router B

Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
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	

Hierarchical Routing

▶ Hierarchical routing reduces the work of route computation but may result in slightly longer paths than flat routing
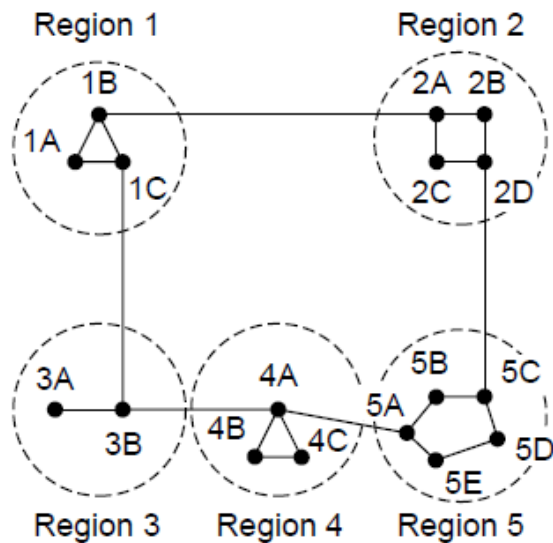
Full table for 1A

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

Best choice to reach nodes in 5 except for 5C

Broadcast Routing

- Broadcast sends a packet to all nodes
    - RPF (Reverse Path Forwarding): send broadcast received on the link to the source out all remaining links
    - Alternatively, can build and use sink trees at all nodes
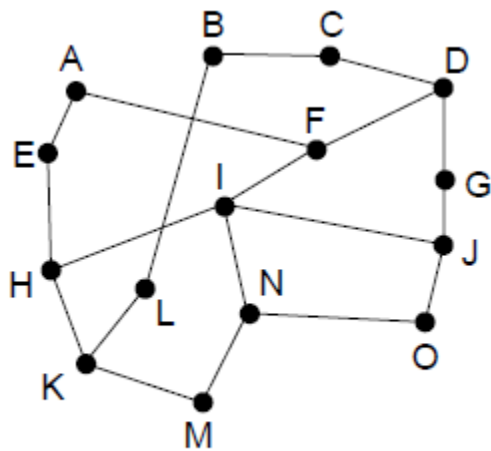
Network
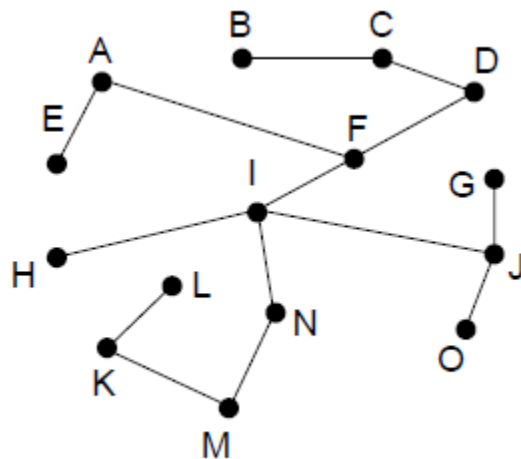
Sink tree for *I* is efficient broadcast
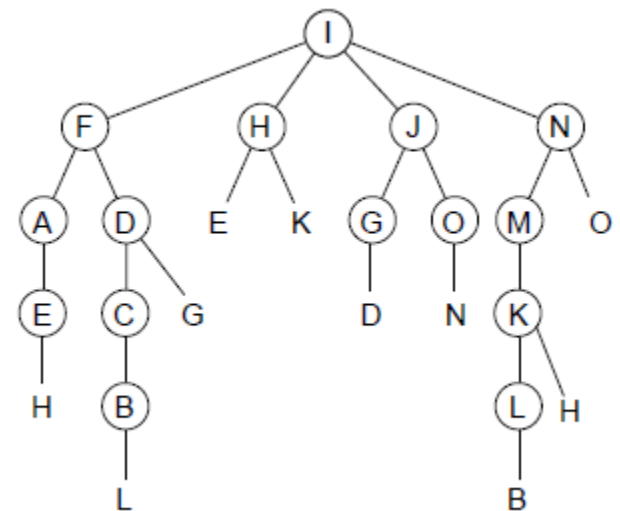
RPF from *I* is larger than sink tree

Multicast Routing (1) – Dense Case

▶ <u>Multicast</u> sends to a subset of the nodes called a group

  ▶ Uses a different tree for each group and source
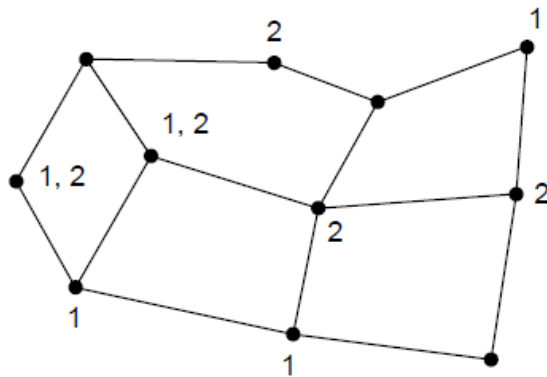
Network with groups 1 & 2
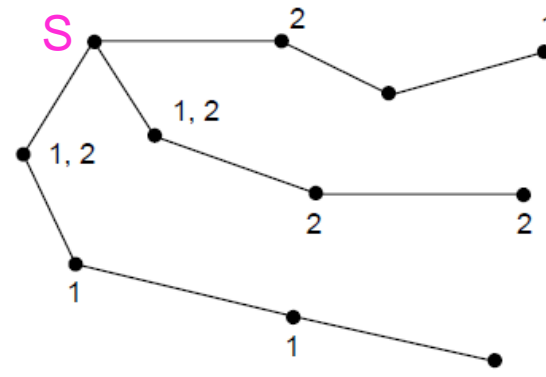
Spanning tree from source S
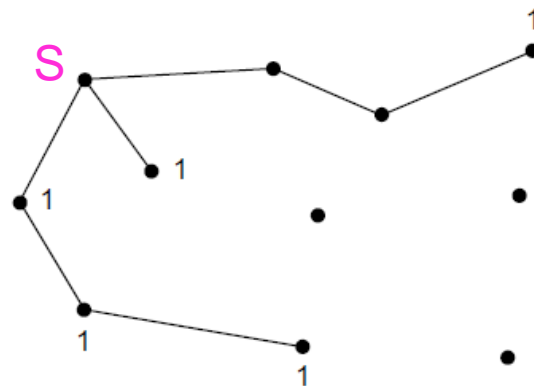
Multicast tree from S to group 1
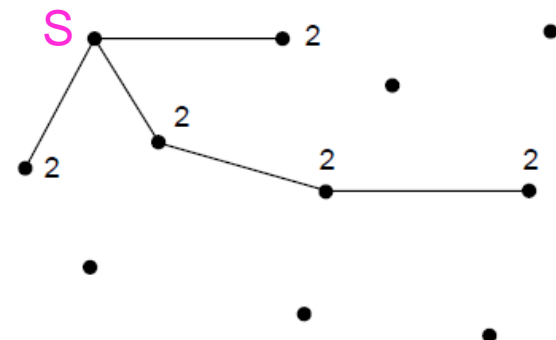
Multicast tree from S to group 2

Multicast Routing (2) – Sparse Case

▶ CBT (Core-Based Tree) uses a single tree to multicast

   ▶ Tree is the sink tree from core node to group members

   ▶ Multicast heads to the core until it reaches the CBT

▶ p 1.
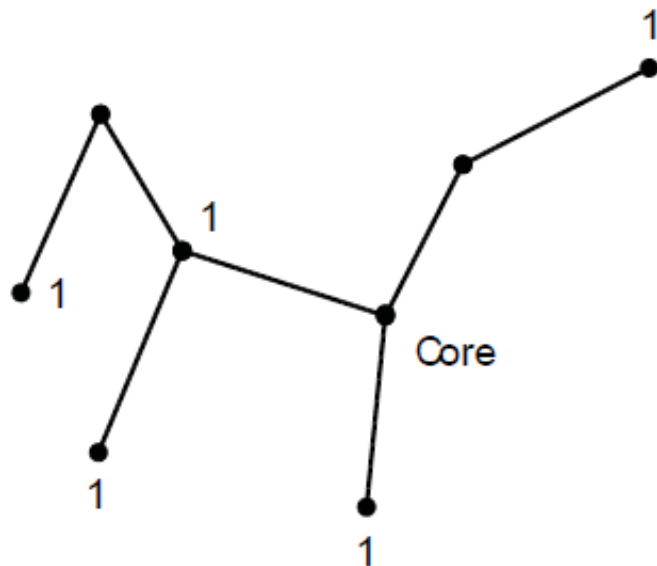
Sink tree from core to group 1
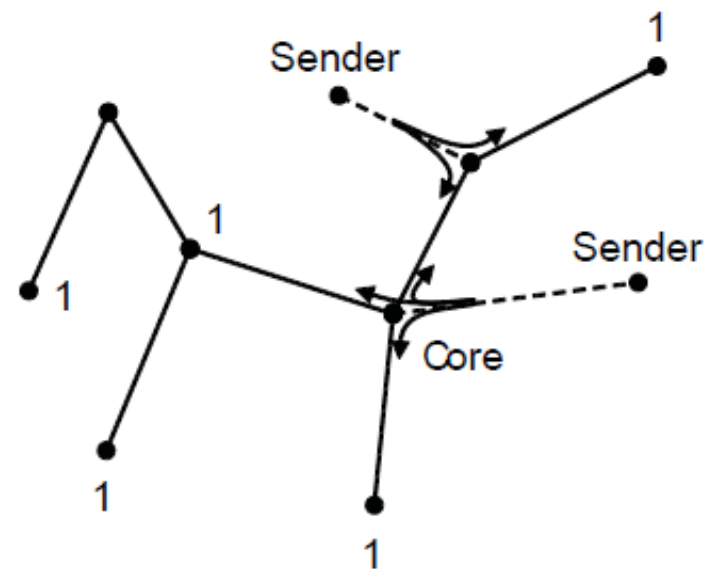
Multicast is send to the core then down when it reaches the sink tree

Anycast Routing

Anycast sends a packet to one (nearest) group member

▶ Falls out of regular routing with a node in many places
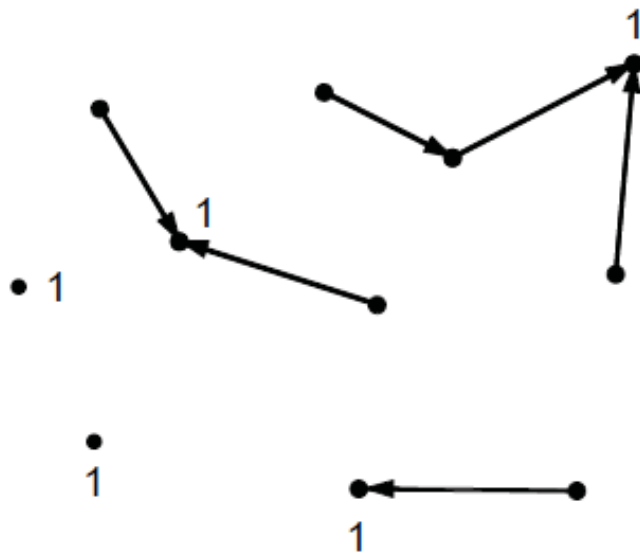
Anycast routes to group 1
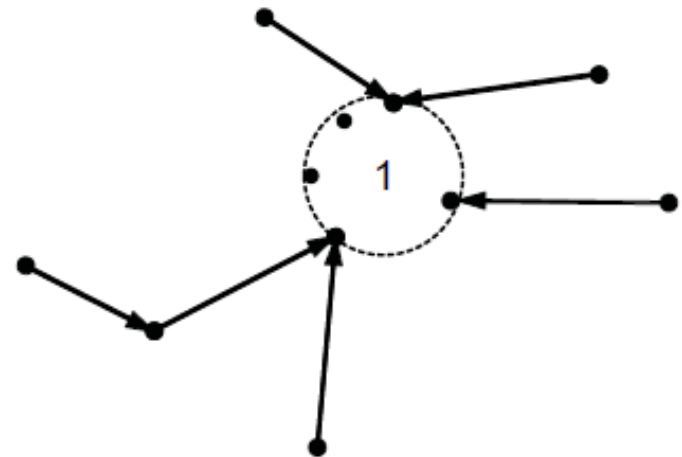
Apparent topology of sink tree to "node" 1

Routing for Mobile Hosts

▶ Mobile hosts can be reached via a home agent

  ▶ Fixed home agent tunnels packets to reach the mobile host; reply can optimize path for subsequent packets

  ▶ No changes to routers or fixed hosts

Sender

2: Send to home address

4: Reply to sender

5: Tunnel to care of address

1: Register care of address

3: Tunnel to care of address

Home agent at home address

Mobile host at care of address
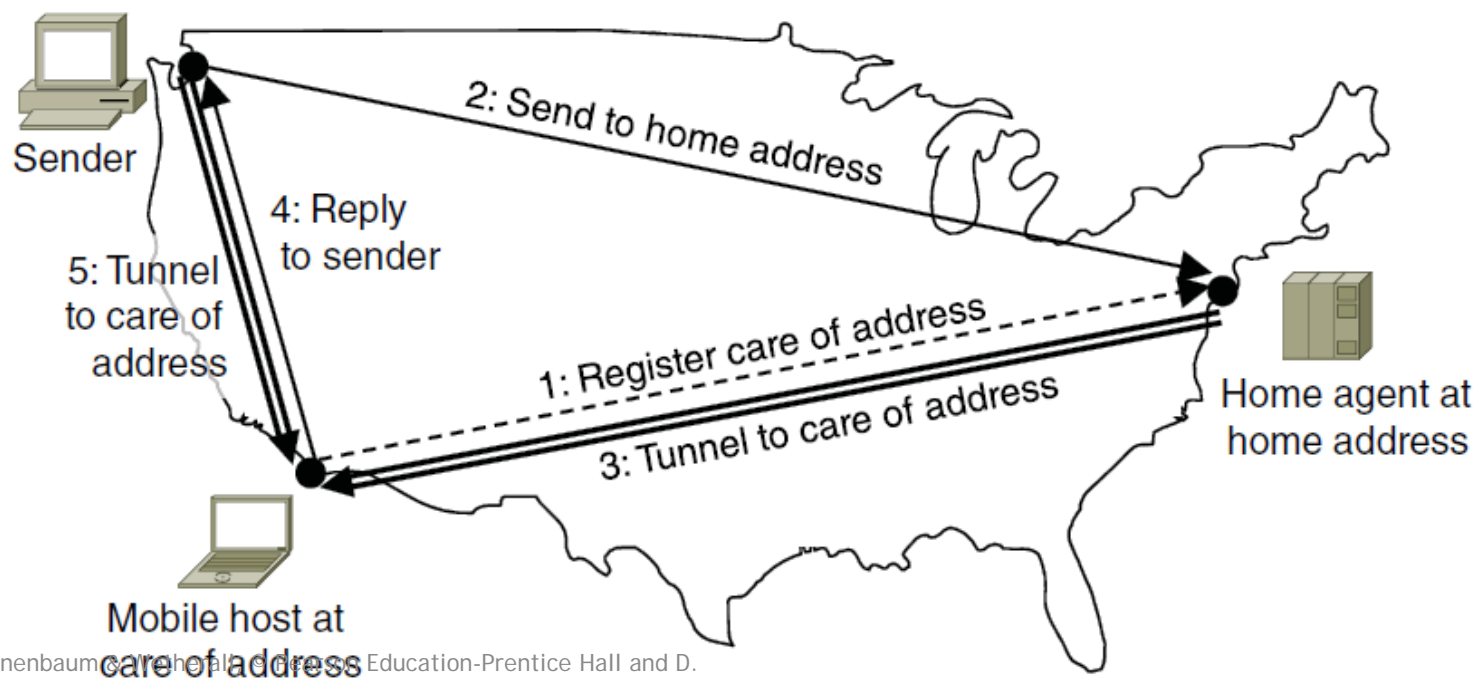
Routing in Ad Hoc Networks

▶ The network topology changes as wireless nodes move

  ▶ Routes are often made on demand, e.g., AODV (below)
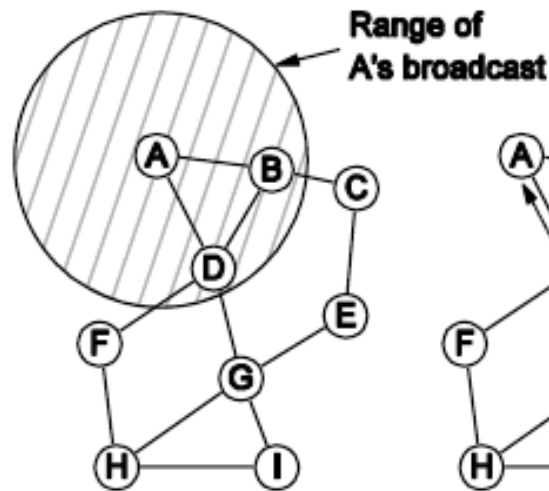
A's starts to find route to *I*
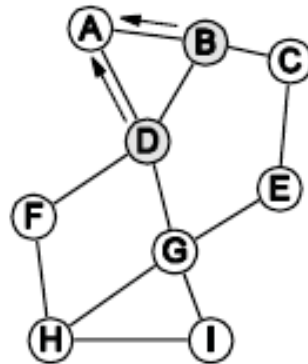
A's broadcast reaches *B* & *D*
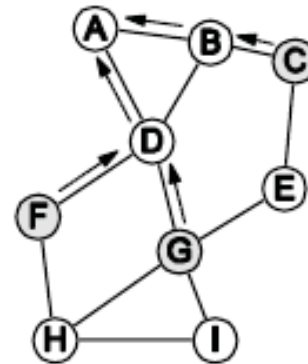
B's and *D*'s broadcast reach *C*, *F* & *G*
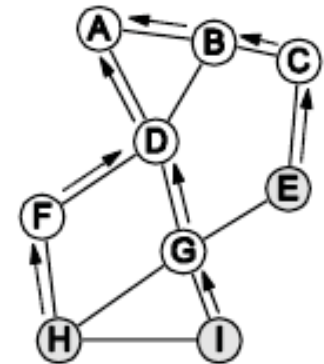
C's, *F*'s and *G*'s broadcast reach *H* & *I*

Congestion Control (1)

Handling congestion is the responsibility of the Network and Transport layers working together

▶ We look at the Network portion here

▶ Traffic-aware routing »

▶ Admission control »

▶ Traffic throttling »

▶ Load shedding »

Congestion Control (2)

Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions

► Goodput (=useful packets) trails offered load

CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011
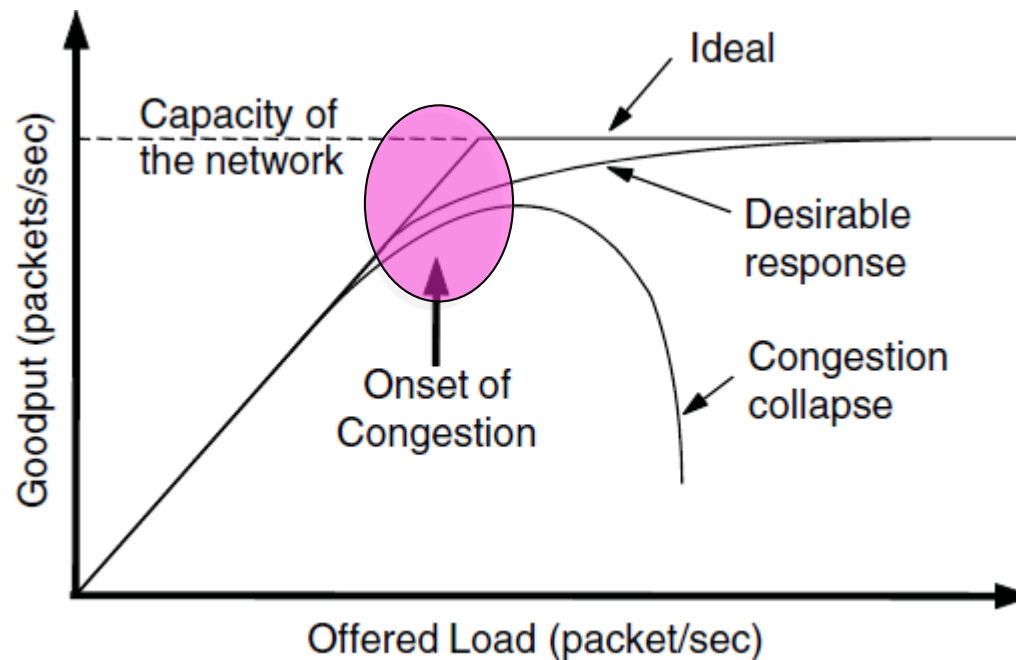
Congestion Control (3) – Approaches

Network must do its best with the offered load

- ▶ Different approaches at different timescales
- ▶ Nodes should also reduce offered load (Transport)

Network provisioning	Traffic-aware routing	Admission control	Traffic throttling	Load shedding

Slower (Preventative) ◀————————————————————————▶ Faster (Reactive)
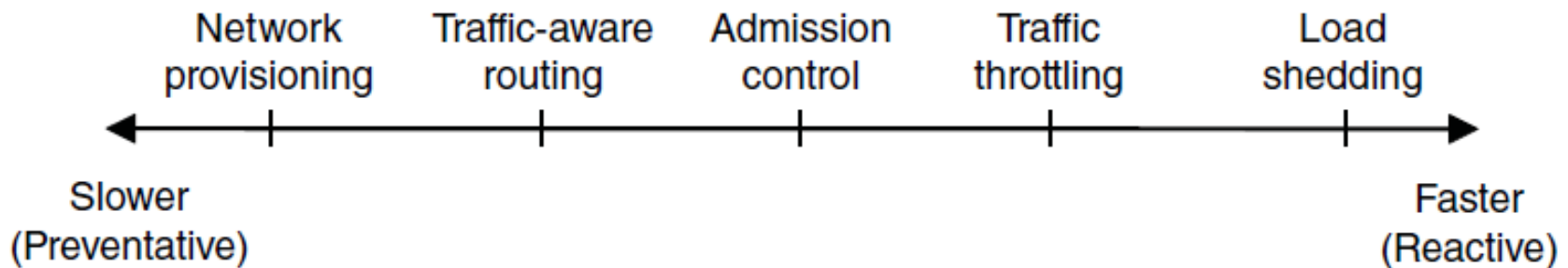
Traffic-Aware Routing

Choose routes depending on traffic, not just topology

▶ E.g., use *EI* for West-to-East traffic if *CF* is loaded

▶ But take care to avoid oscillations
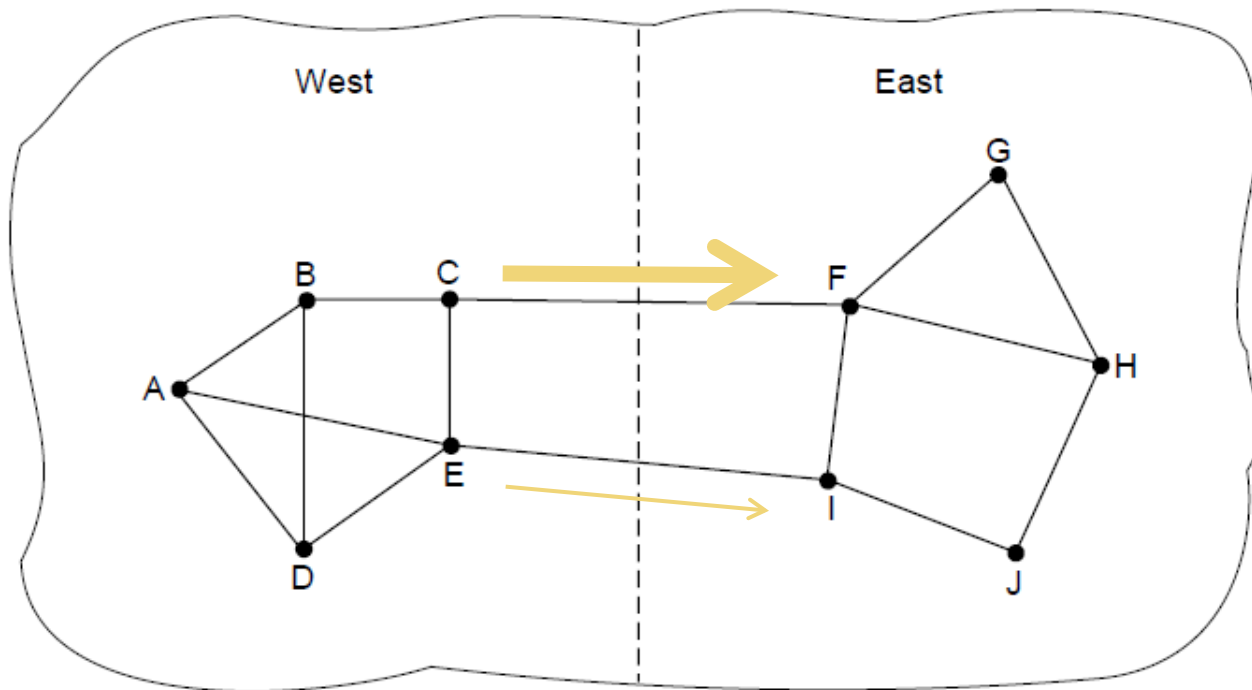
Admission Control

▶ Admission control allows a new traffic load only if the network has sufficient capacity, e.g., with virtual circuits

   ▶ Can combine with looking for an uncongested route
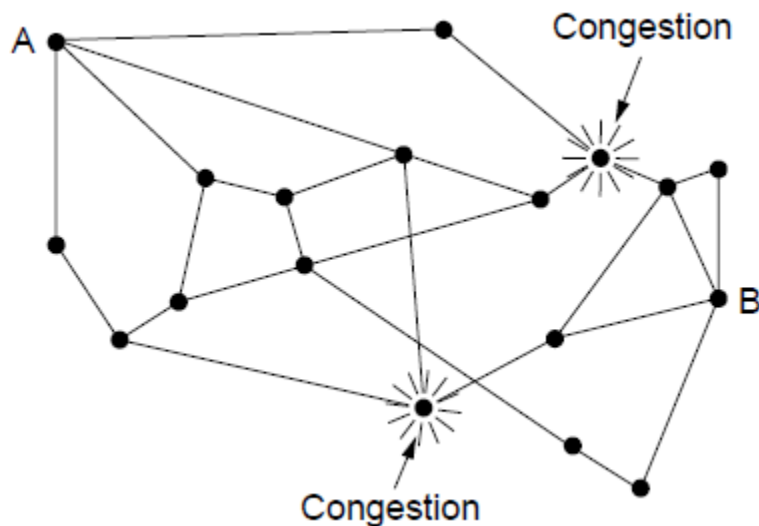
Network with some
congested nodes
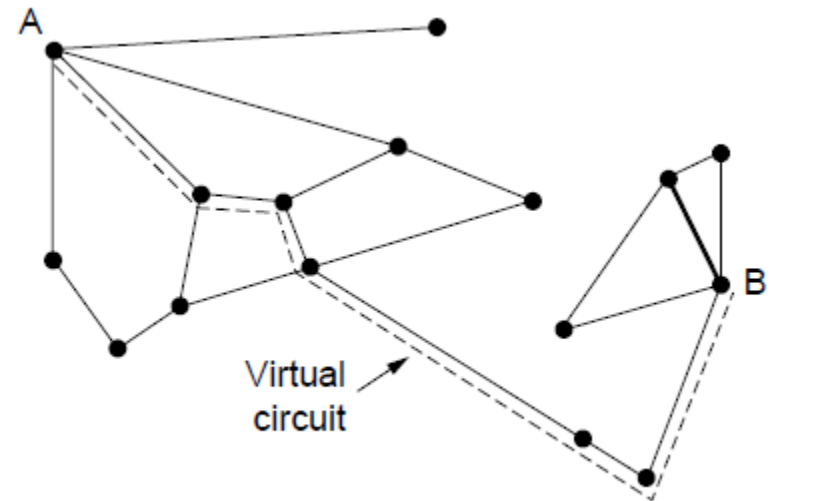
Uncongested portion and
route *AB* around congestion

Traffic Throttling

Congested routers signal hosts to slow down traffic

▶ ECN (Explicit Congestion Notification) marks packets and receiver returns signal to sender
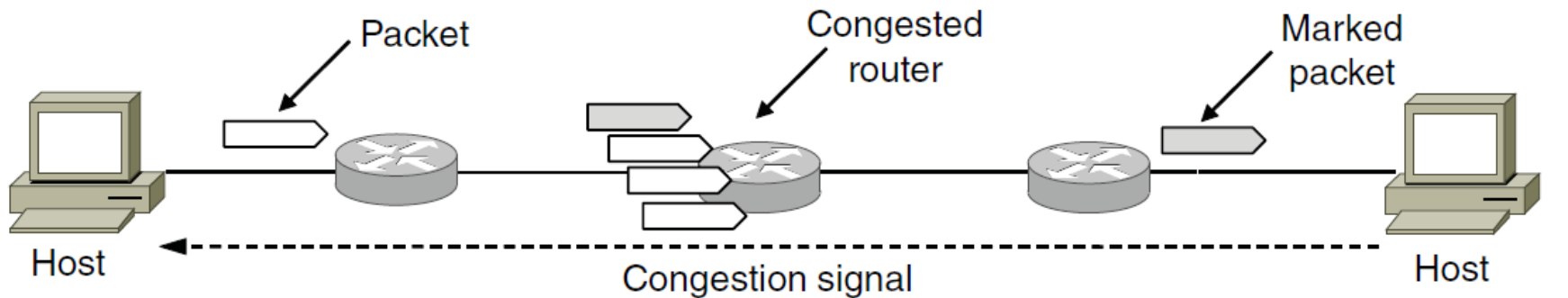
Load Shedding (1)

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
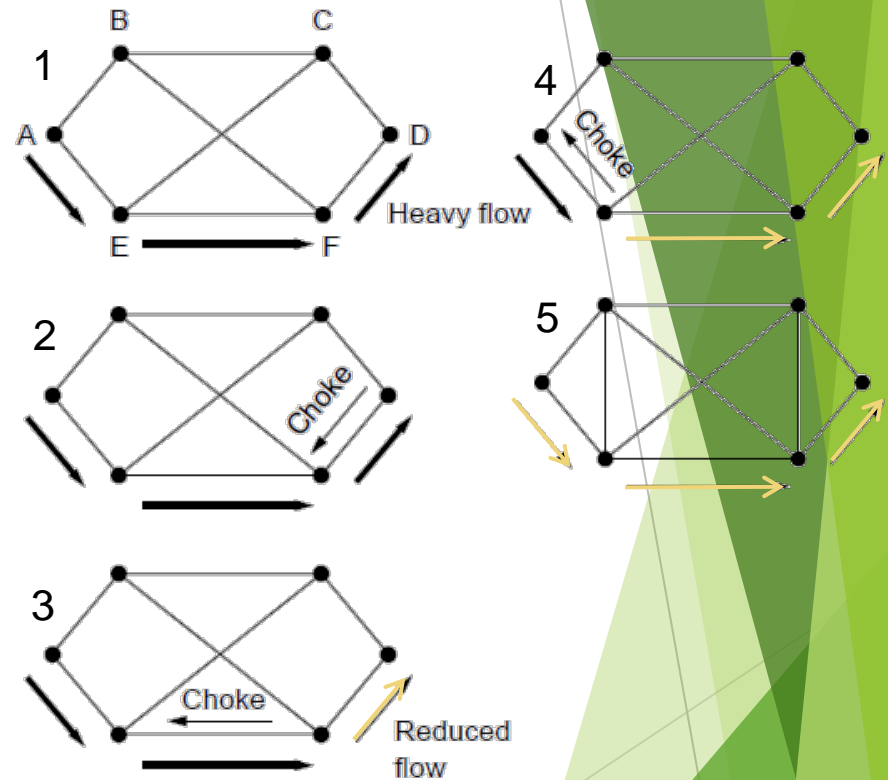
Load Shedding (2)

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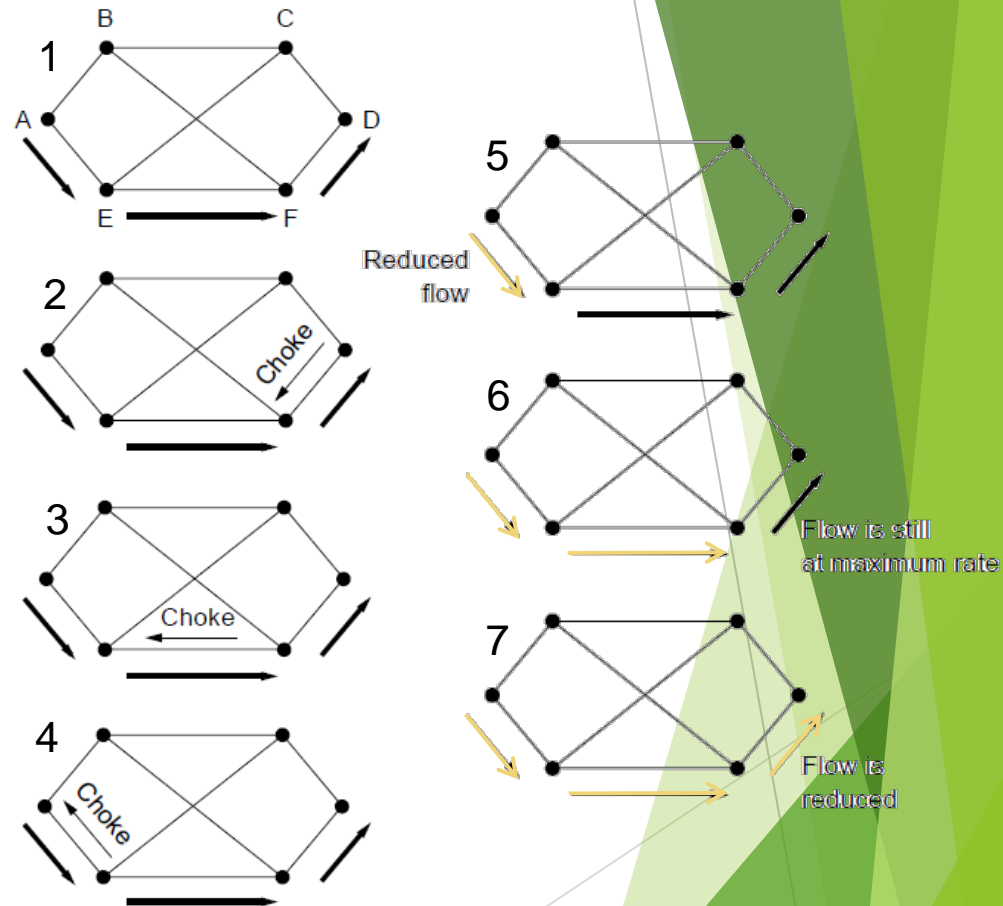
Quality of Service

- ▶ Application requirements »
- ▶ Traffic shaping »
- ▶ Packet scheduling »
- ▶ Admission control »
- ▶ Integrated services »
- ▶ Differentiated services »

Application Requirements (1)

Different applications care about different properties

► We want all applications to get what they need

.

Application	Bandwidth	Delay	Jitter	Loss
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

"High" means a demanding requirement, e.g., low delay

Application Requirements (2)

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

Network Service	Application
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 (1)

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

▶ Lets us make guarantees
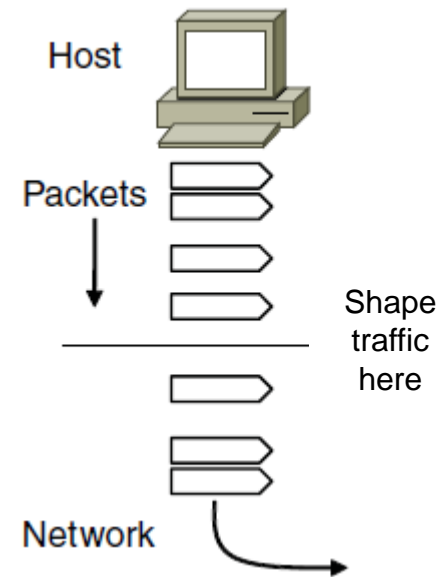
Traffic Shaping (2)

Token/Leaky bucket limits both the average rate (R) and short-term burst (B) of traffic

▶ For token, bucket size is B, water enters at rate R and is removed to send; opposite for leaky.
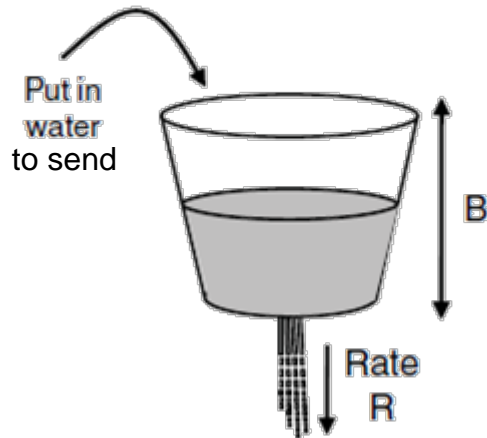
Leaky bucket
(need not full to send)
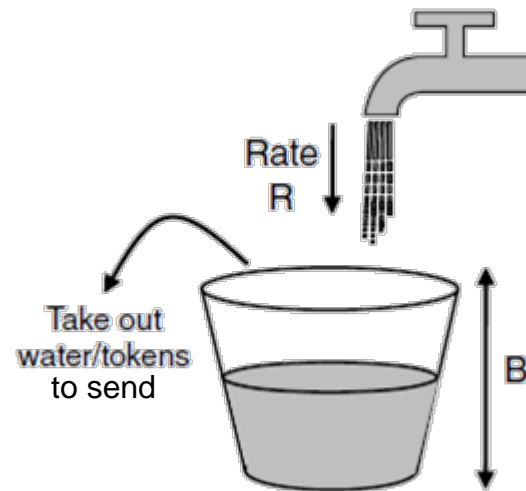
Token bucket
(need some water to send)

Traffic Shaping (3)

Host traffic
R=200 Mbps
B=16000 KB

↓

Shaped by
R=200 Mbps
B=9600 KB

↓

Shaped by
R=200 Mbps
B=0 KB

Rate (Mbps)

1000 — 125 MB/s for 125msec

25 MB/s for 250msec

(a)

With R=25 MB/s, B=9600 KB

(b)

With R=25 MB/s, B=0

Time (msec)   1000

(c)

Bucket (KB)

16000

(d)

9600 — Bucket empties, traffic delayed

(e)

Bucket always empty

0

Time (msec)   1000

(f)
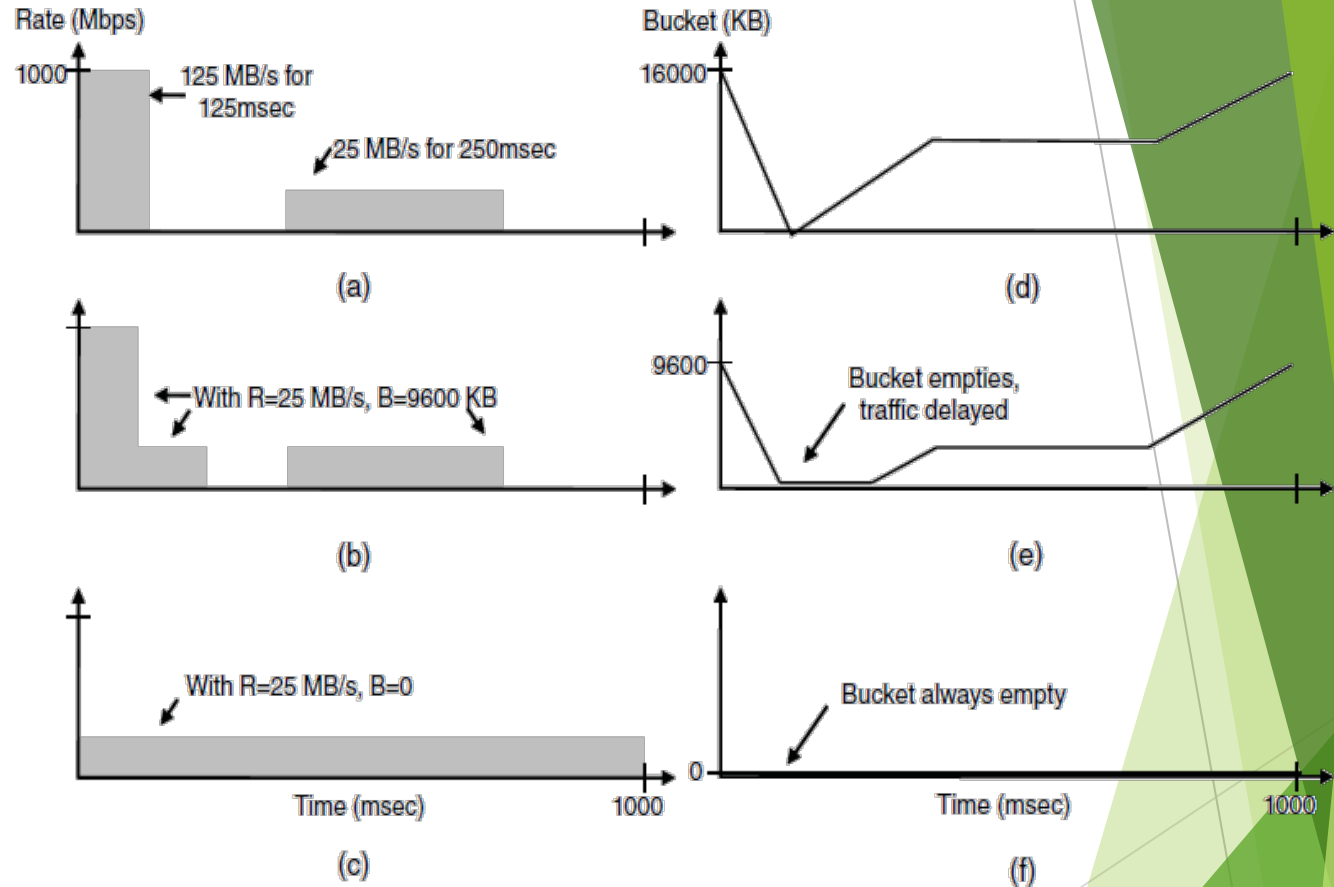
Smaller bucket size delays traffic and reduces burstiness

Packet Scheduling (1)

Packet scheduling divides router/link resources among traffic flows with alternatives to FIFO (First In First Out)
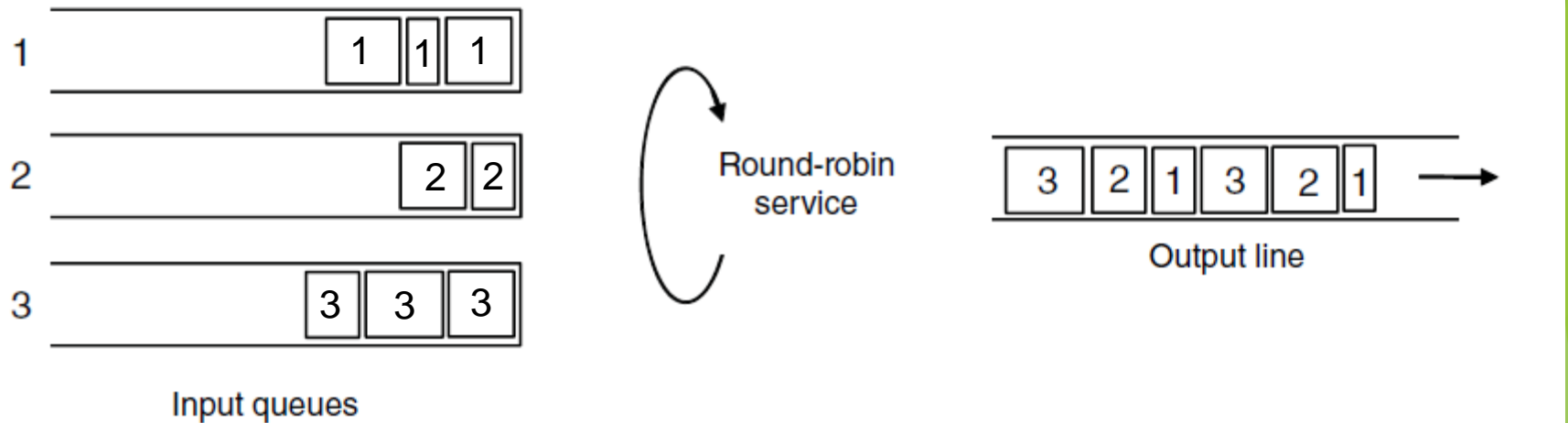
Example of round-robin queuing

Packet Scheduling (2)

Fair Queueing approximates bit-level fairness with different packet sizes; weights change target levels

▶ Result is WFQ (Weighted Fair Queueing)

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

$$F_i = \max(A_i, F_{i-1}) + L_i/W$$
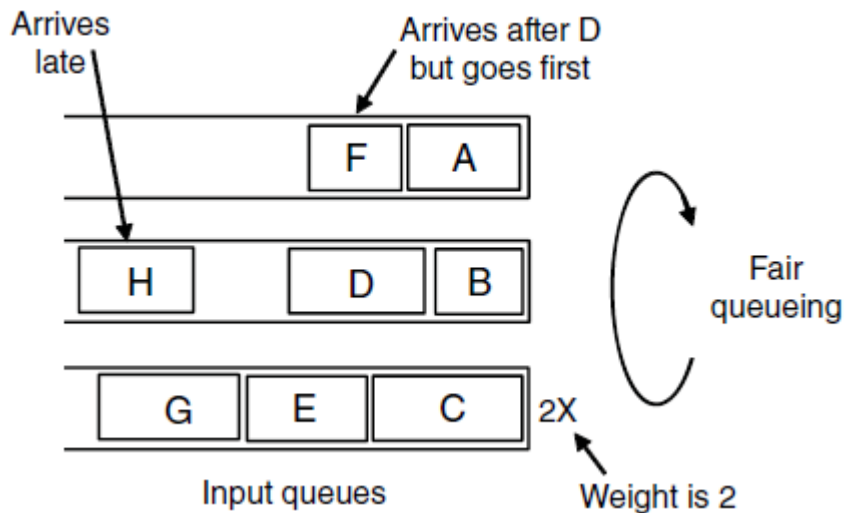
Packets may be sent out of arrival order

Finish virtual times determine transmission order

Admission Control (1)

Admission control takes a traffic flow specification and decides whether the network can carry it

▶ Sets up packet scheduling to meet QoS

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

Example flow specification

Admission Control (2)

Construction to guarantee bandwidth B and delay D:

▶ Shape traffic source to a (R, B) token bucket

▶ Run WFQ with weight W / all weights > R/capacity

▶ Holds for all traffic patterns, all topologies
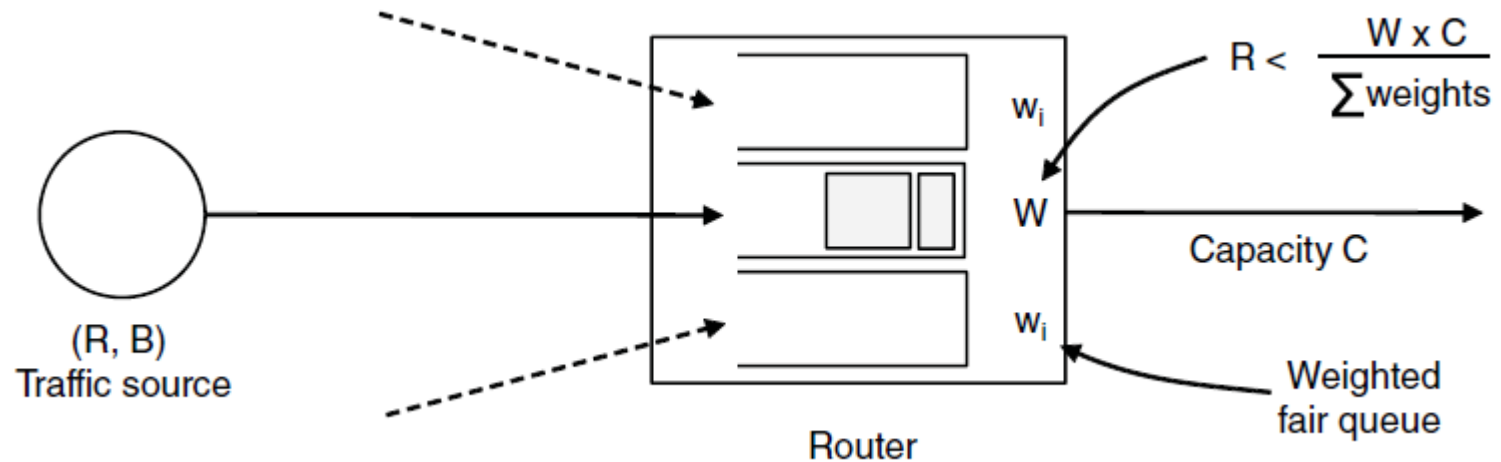
Integrated Services (1)

Design with QoS for each flow; handles multicast traffic.

Admission with RSVP (Resource reSerVation Protocol):

- ▶ Receiver sends a request back to the sender
- ▶ Each router along the way reserves resources
- ▶ Routers merge multiple requests for same flow
- ▶ Entire path is set up, or reservation not made

Integrated Services (2)
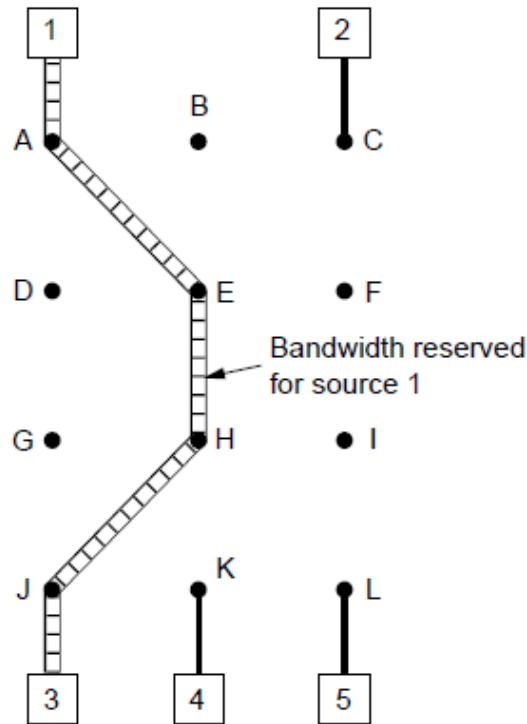
R3 reserves flow from S1
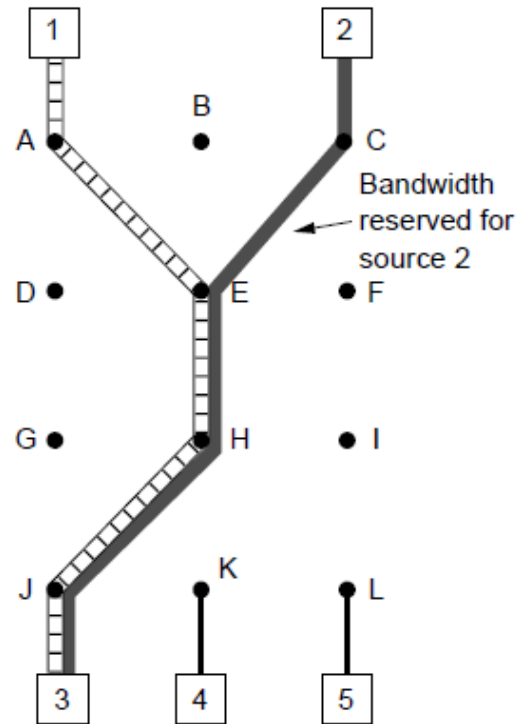
R3 reserves flow from S2
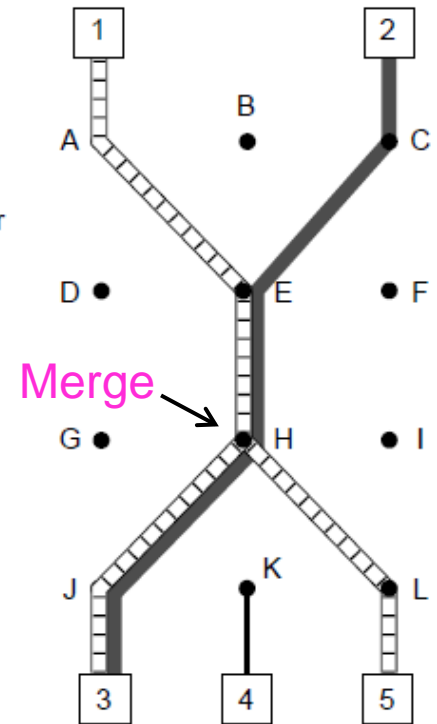
R5 reserves flow from S1; merged with R3 at H

Differentiated Services (1)

- Design with classes of QoS; customers buy what they want
  - Expedited class is sent in preference to regular class
  - Less expedited traffic but better quality for applications
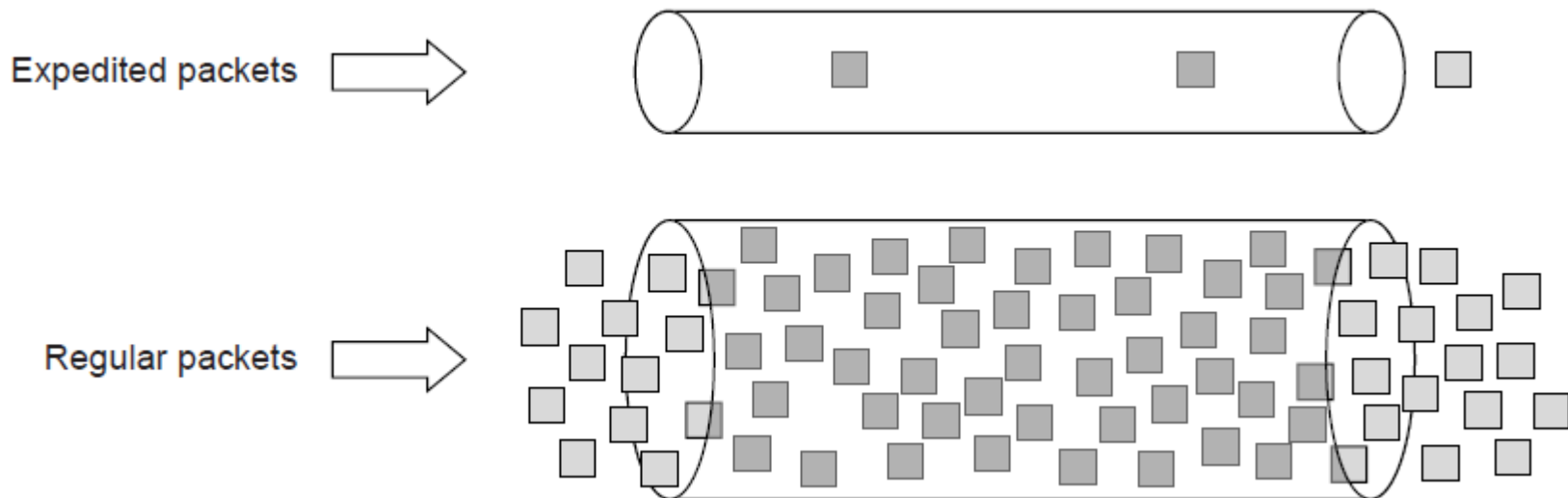
Differentiated Services (2)

Implementation of DiffServ:

▶ Customers mark desired class on packet

▶ ISP shapes traffic to ensure markings are paid for

▶ Routers use WFQ to give different service levels
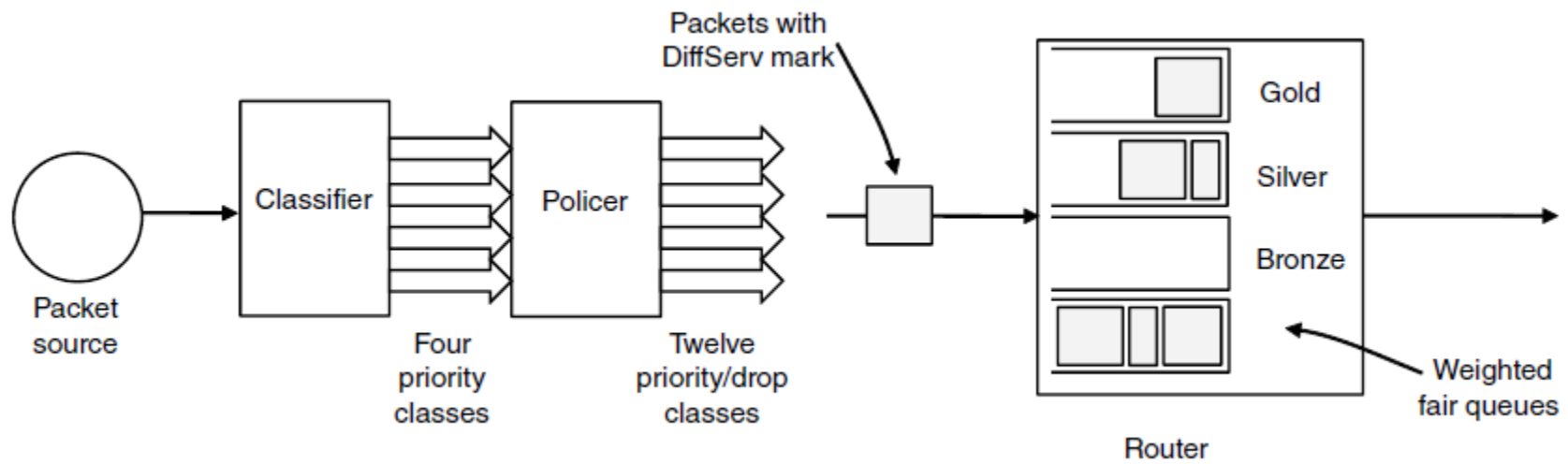
Internetworking

Internetworking joins multiple, different networks into a single larger network

- ► How networks differ »
- ► How networks can be connected »
- ► Tunneling »
- ► Internetwork routing »
- ► Packet fragmentation »

How Networks Differ

Differences can be large; complicates internetworking

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

Internetworking based on a common network layer – IP

Packet mapped to a VC here

Common protocol (IP) carried all the way
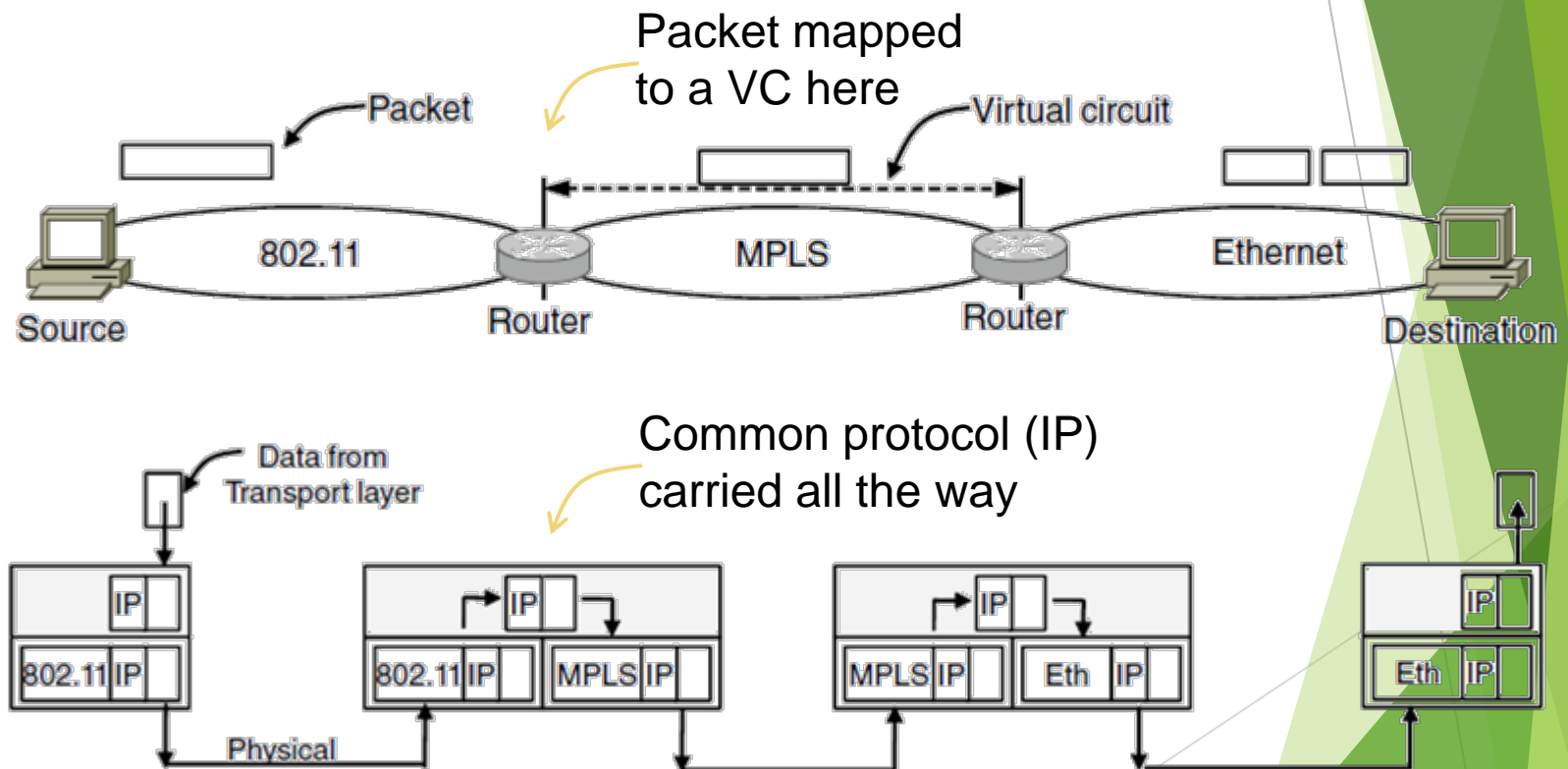
Tunneling (1)

Connects two networks through a middle one

▶ Packets are encapsulates over the middle
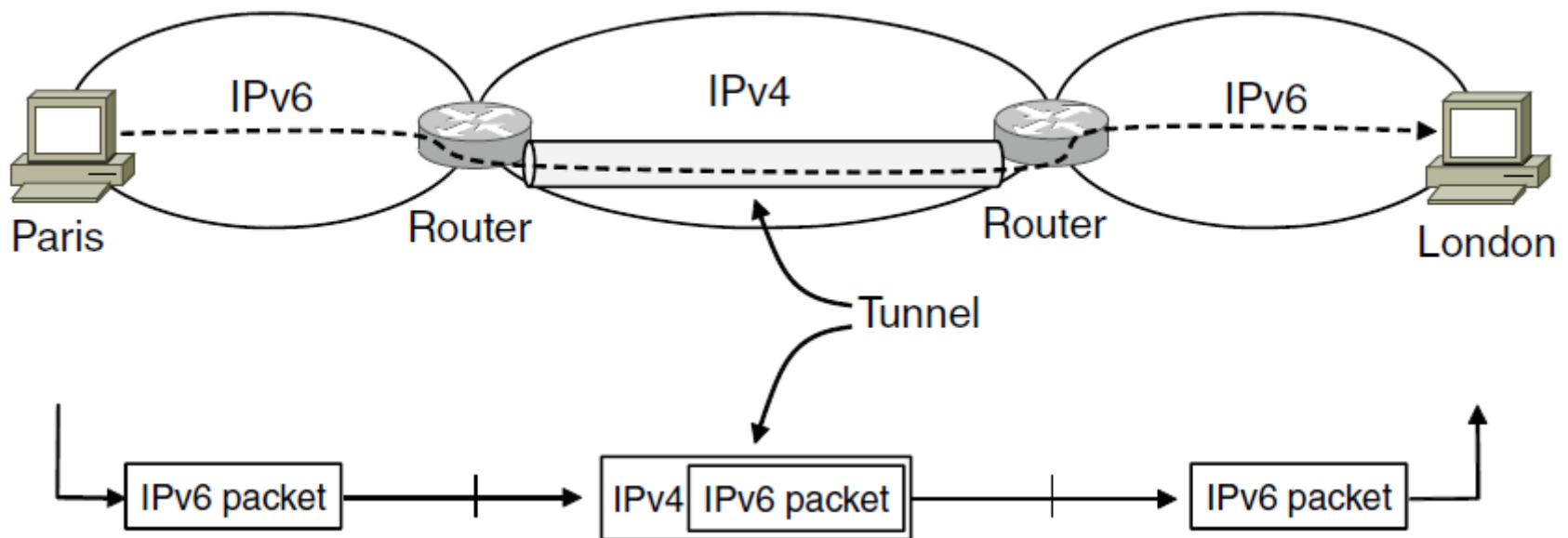
Tunneling (2)

Tunneling analogy:

- ▶ tunnel is a link; packet can only enter/exit at ends
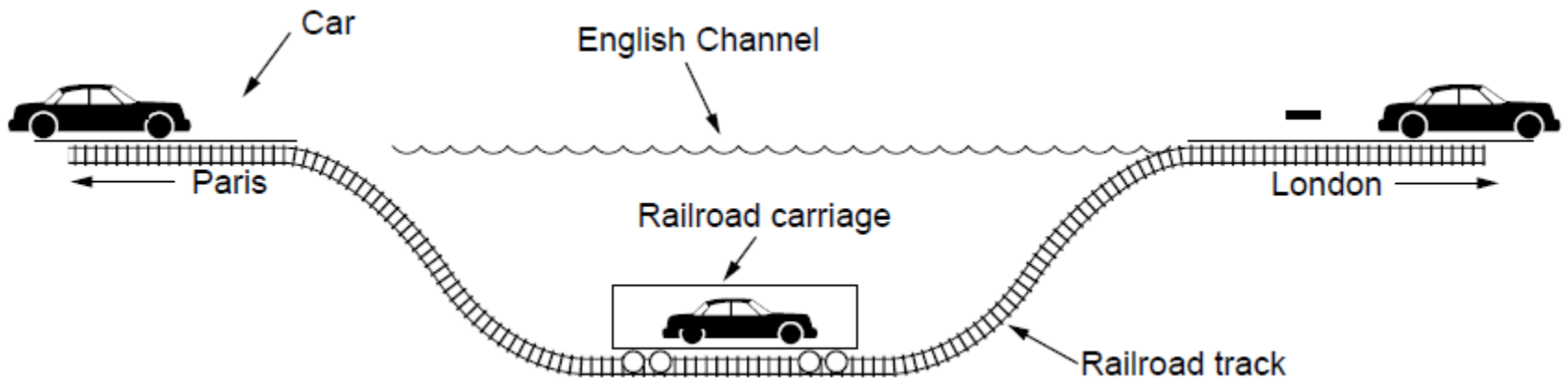
Packet Fragmentation (1)

▶ Networks have different packet size limits for many reasons

   ▶ Large packets sent with fragmentation & reassembly

G_1 fragments G_2 reassembles G_3 fragments G_4 reassembles
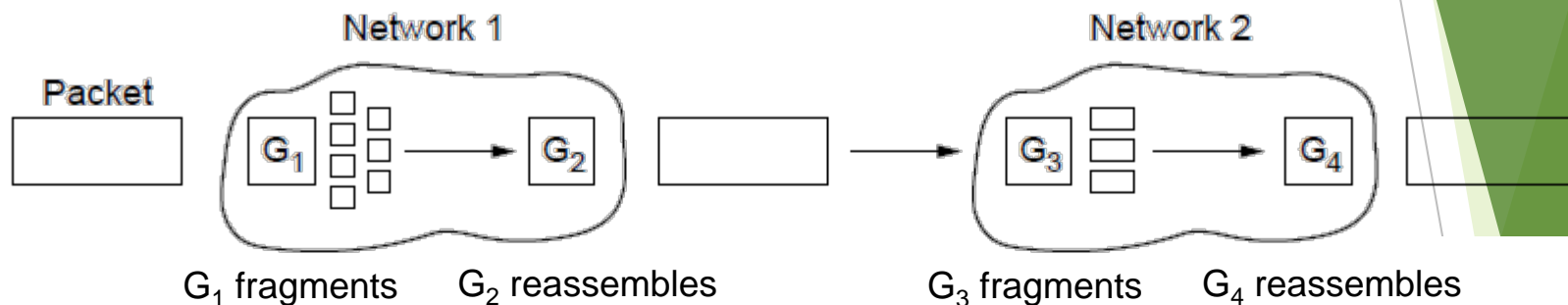
Transparent – packets fragmented / reassembled in each network

G_1 fragments

… destination will reassemble
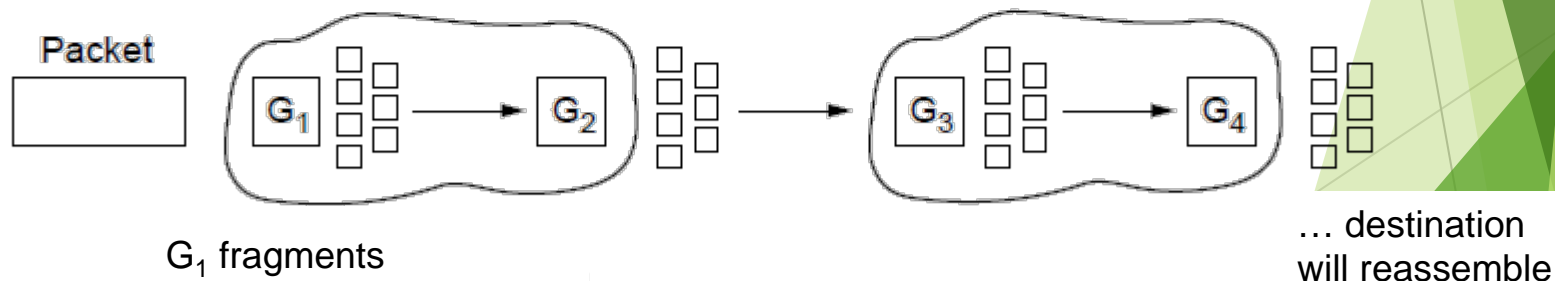
Non-transparent – fragments are reassembled at destination

Packet Fragmentation (2)

▶ Example of IP-style fragmentation:

Original packet: (10 data bytes)
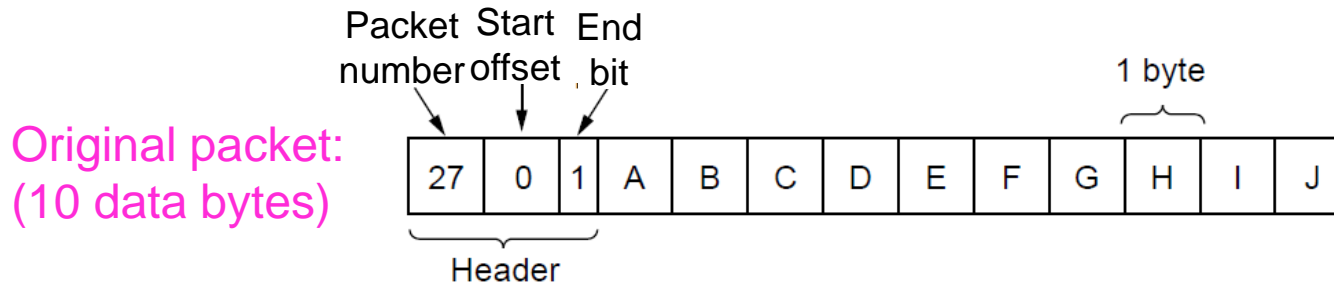
Fragmented: (to 8 data bytes)
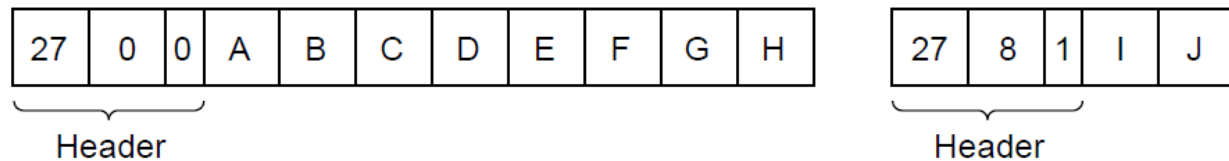
Re-fragmented: (to 5 bytes)
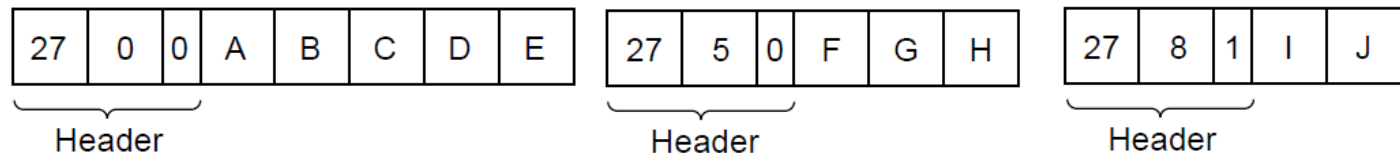
Packet Fragmentation (3)

Path MTU Discovery avoids network fragmentation

▶ Routers return MTU (Max. Transmission Unit) to source and discard large packets
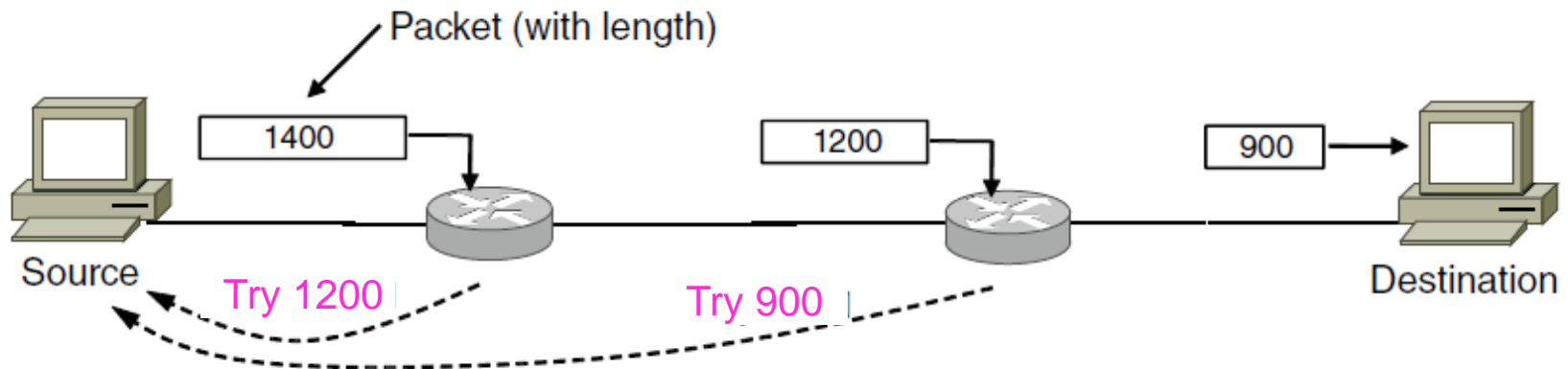
Network Layer in the Internet (1)

- ▶ IP Version 4 »
- ▶ IP Addresses »
- ▶ IP Version 6 »
- ▶ Internet Control Protocols »
- ▶ Label Switching and MPLS »
- ▶ OSPF—An Interior Gateway Routing Protocol »
- ▶ BGP—The Exterior Gateway Routing Protocol »
- ▶ Internet Multicasting »
- ▶ Mobile IP »

Network Layer in the Internet (2)

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

Network Layer in the Internet (3)

► Internet is an interconnected collection of many networks that is held together by the IP protocol
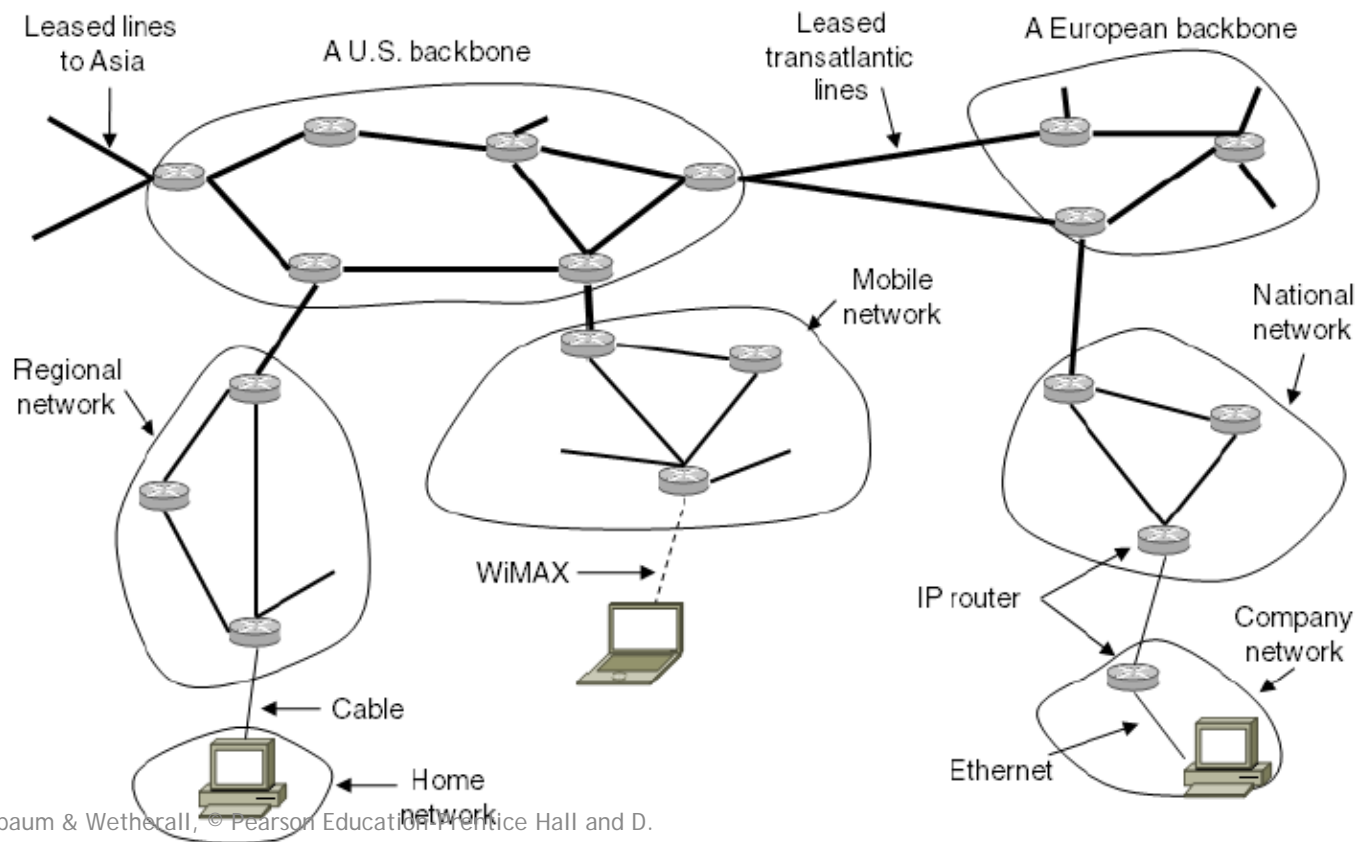
IP Version 4 Protocol (1)

IPv4 (Internet Protocol) header is carried on all packets and has fields for the key parts of the protocol:
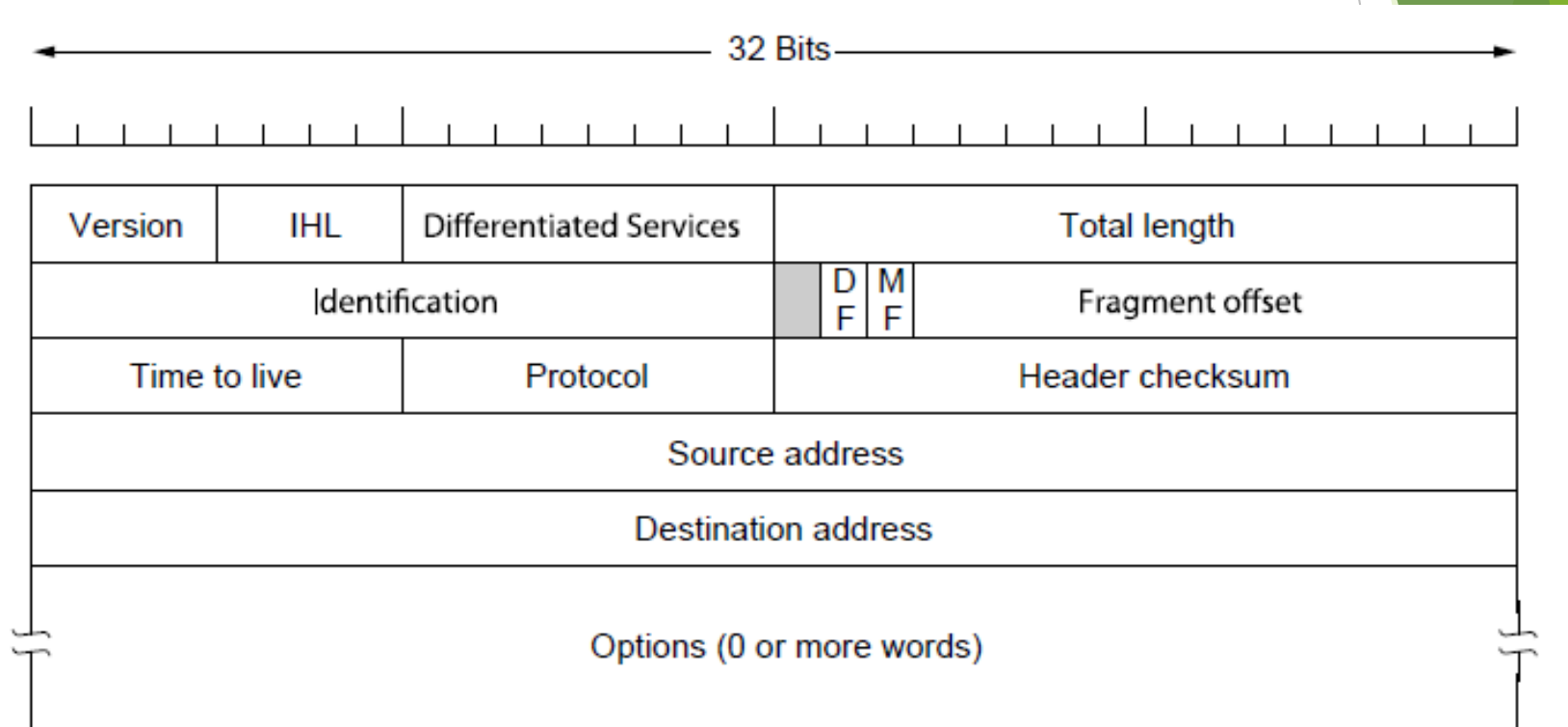
IP Addresses (1) – Prefixes

Addresses are allocated in blocks called <u>prefixes</u>

▶ Prefix is determined by the network portion

▶ Has 2^L addresses aligned on 2^L boundary

▶ Written address/length, e.g., 18.0.31.0/24

```
                          32 bits
|<----------------------------------------------------->|

     Prefix length = L bits              32 -- L bits
|<--------------------------------->|<---------------->|

+-----------------------------------+------------------+
|            Network                |      Host        |
+-----------------------------------+------------------+
Subnet
mask   1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
```
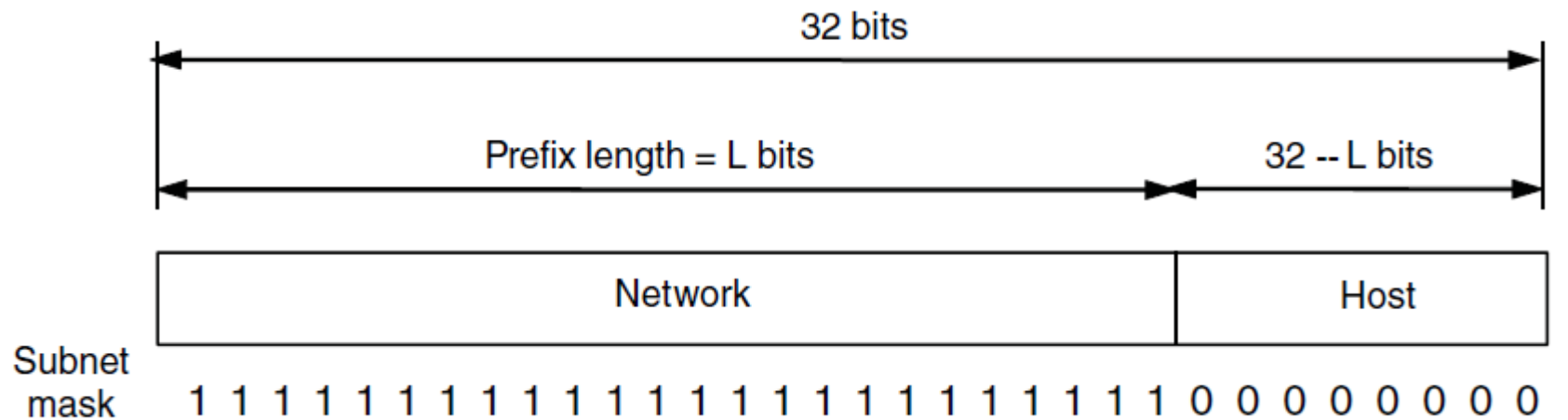
IP Addresses (2) – Subnets

Subnetting splits up IP prefix to help with management

▶ Looks like a single prefix outside the network

EE   128.208.0.0/18

CS   128.208.128.0/17

Art   128.208.96.0/19

128.208.0.0/16
(to Internet)
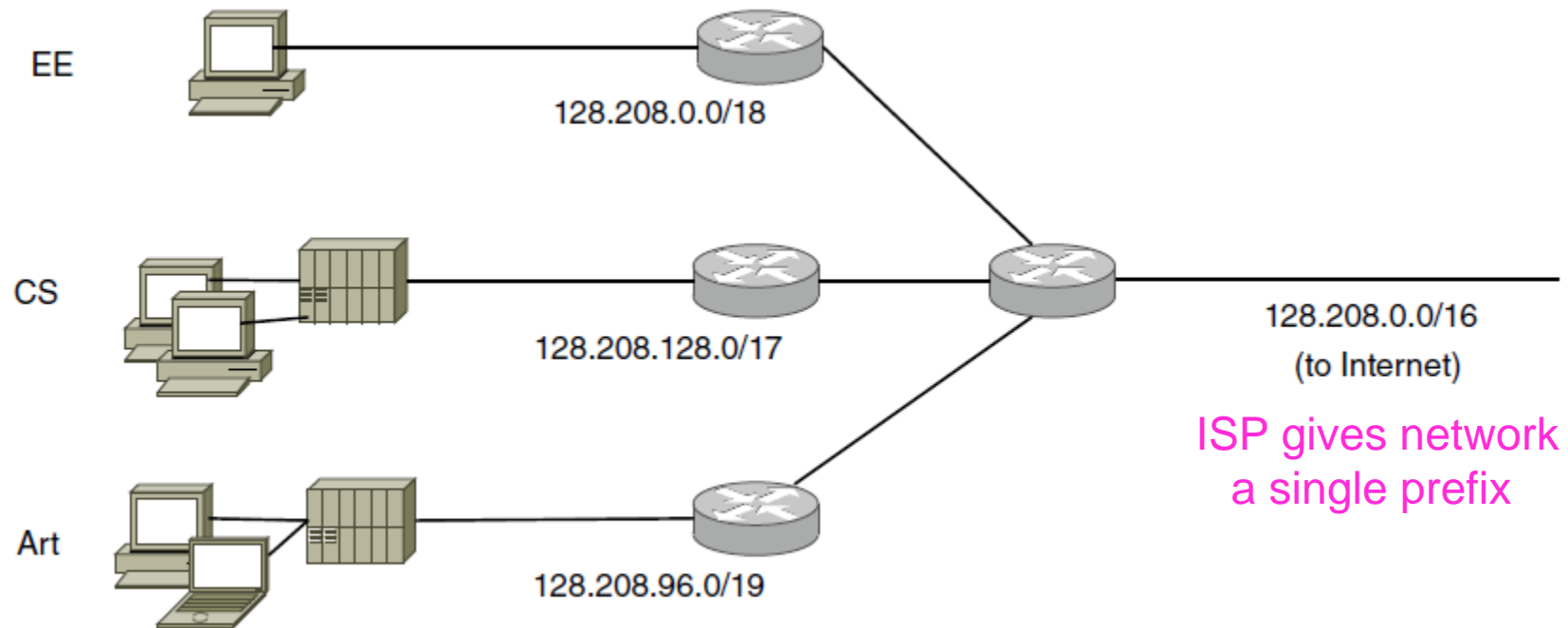
ISP gives network
a single prefix

Network divides it into subnets internally

IP Addresses (3) – Aggregation

Aggregation joins multiple IP prefixes into a single larger prefix to reduce routing table size
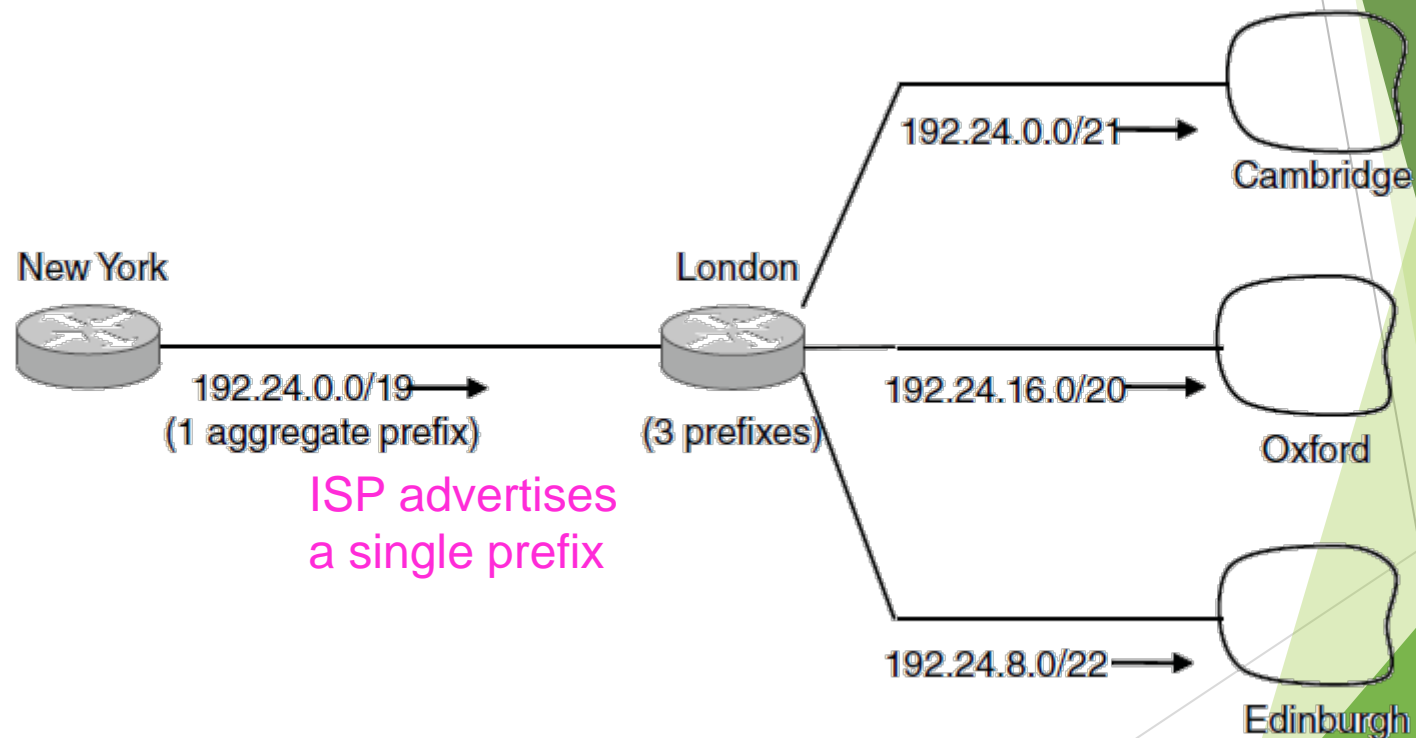
ISP advertises a single prefix

ISP customers have different prefixes

IP Addresses (4) – Longest Matching Prefix

Packets are forwarded to the entry with the <u>longest matching prefix</u> or smallest address block

▶ Complicates forwarding but adds flexibility
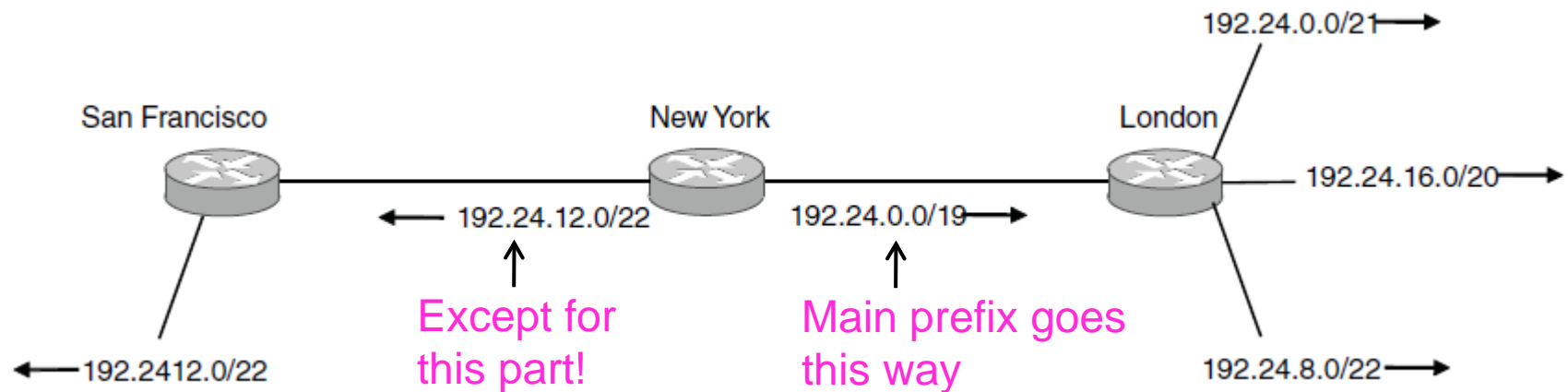
IP Addresses (5) – Classful Addresing

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
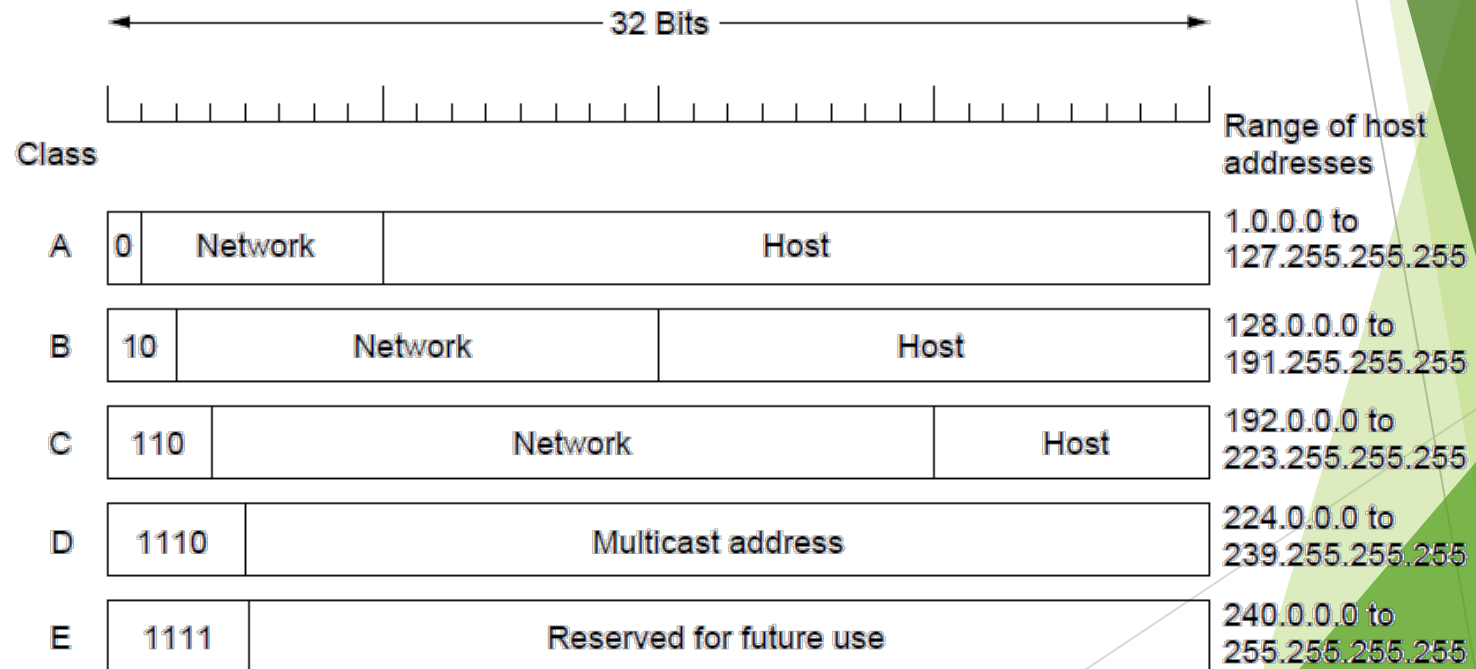
IP Addresses (6) – NAT

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.

Packet before translation

Packet after translation

IP=10.0.0.1
port =5544

IP=198.60.42.12
port =3344

(to Internet)

ISP router

Customer router and LAN

NAT box/firewall

Boundary of customer premises
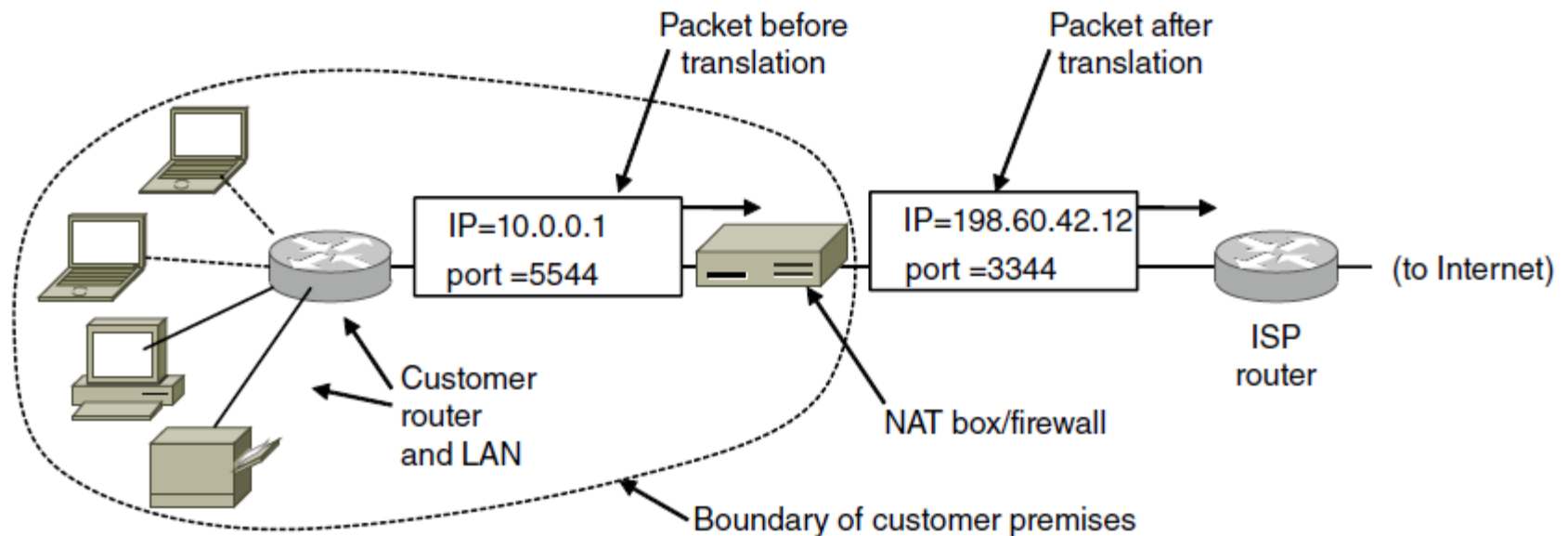
IP Version 6 (1)

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

IP Version 6 (2)

IPv6 protocol header has much longer addresses (128 vs. 32 bits) and is simpler (by using extension headers)
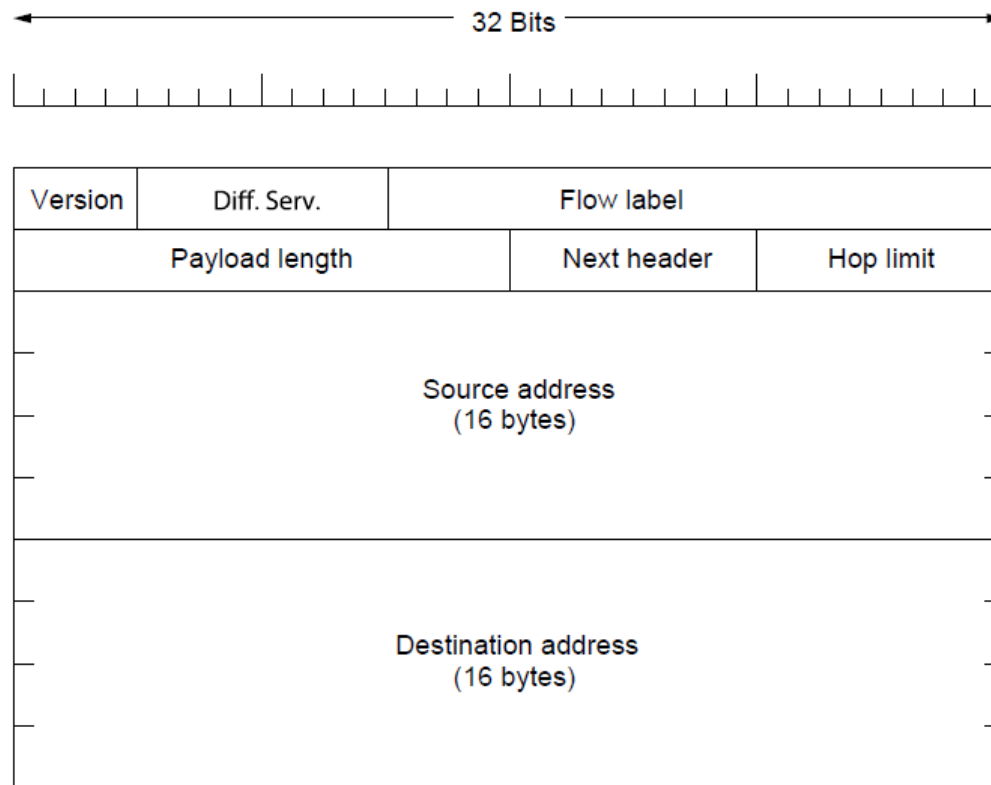
IP Version 6 (3)

IPv6 extension headers handles other functionality

Extension header	Description
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

Internet Control Protocols (1)

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

Internet Control Protocols (2)

Main ICMP (Internet Control Message Protocol) 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

Internet Control Protocols (3)

▶ ARP (Address Resolution Protocol) lets nodes find target Ethernet addresses [pink] from their IP addresses
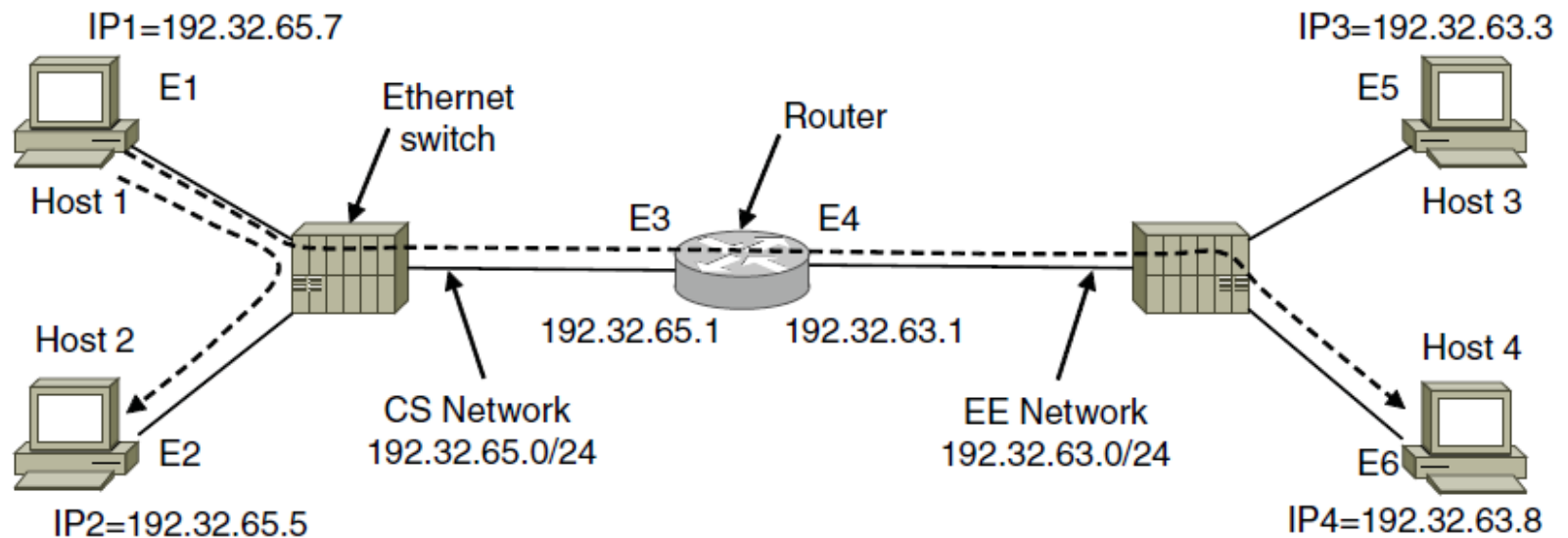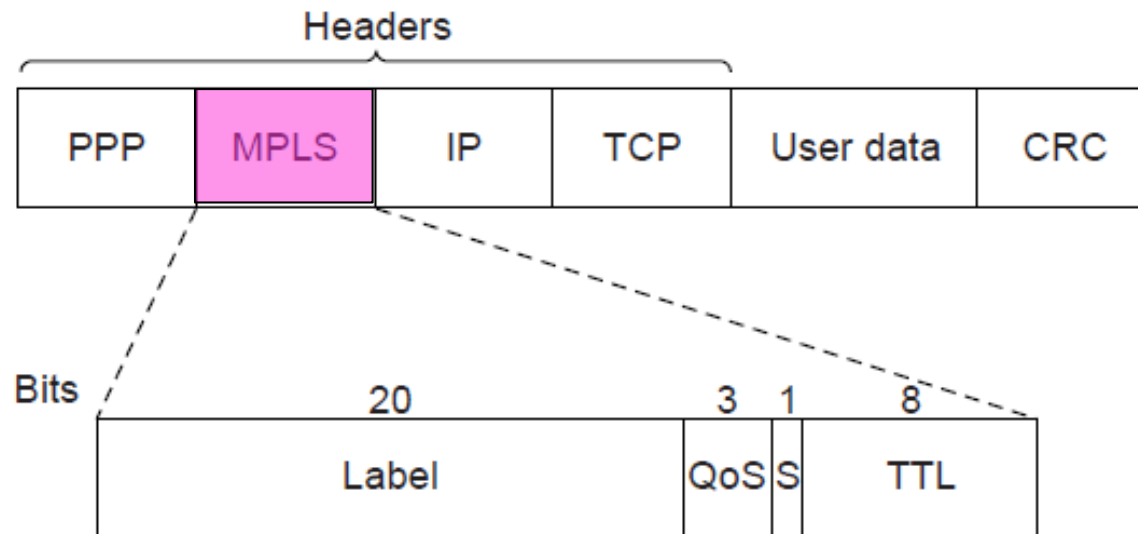

Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Label Switching and MPLS (1)

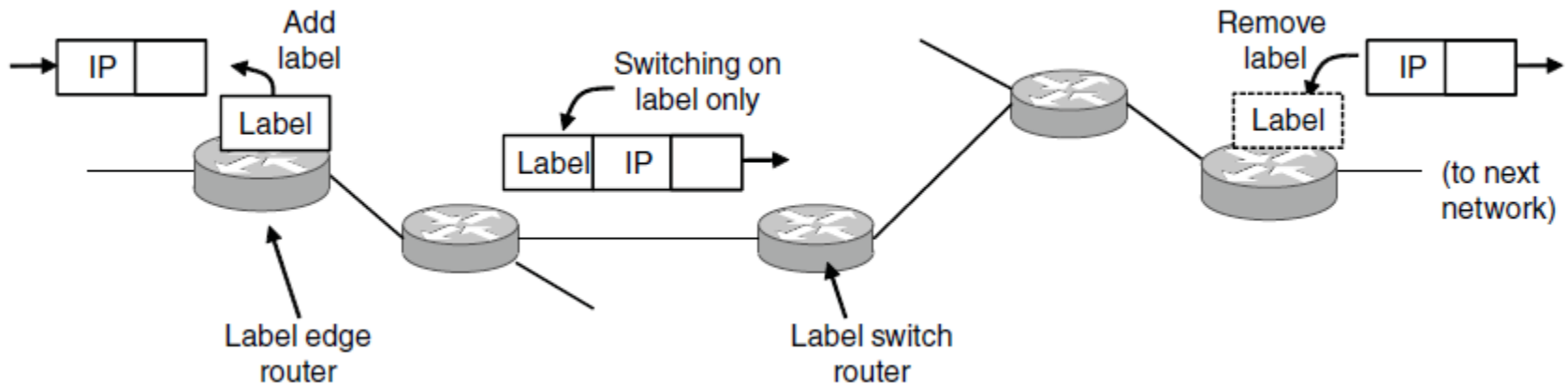MPLS (Multi-Protocol Label Switching) sends packets along established paths; ISPs can use for QoS

▶ Path indicated with label below the IP layer

Label Switching and MPLS (2)

Label added based on IP address on entering an MPLS network (e.g., ISP) and removed when leaving it

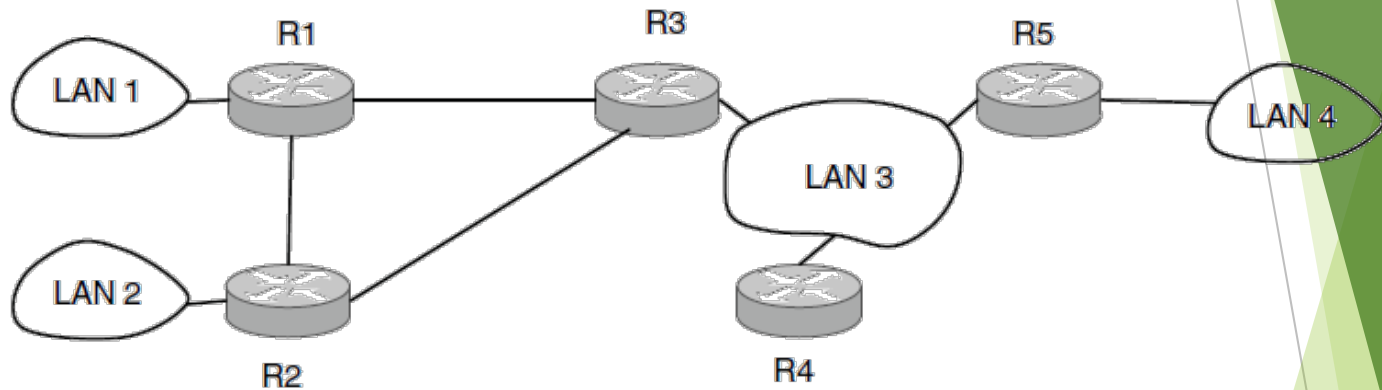▶ Forwarding only uses label inside MPLS network

OSPF— Interior Routing Protocol (1)
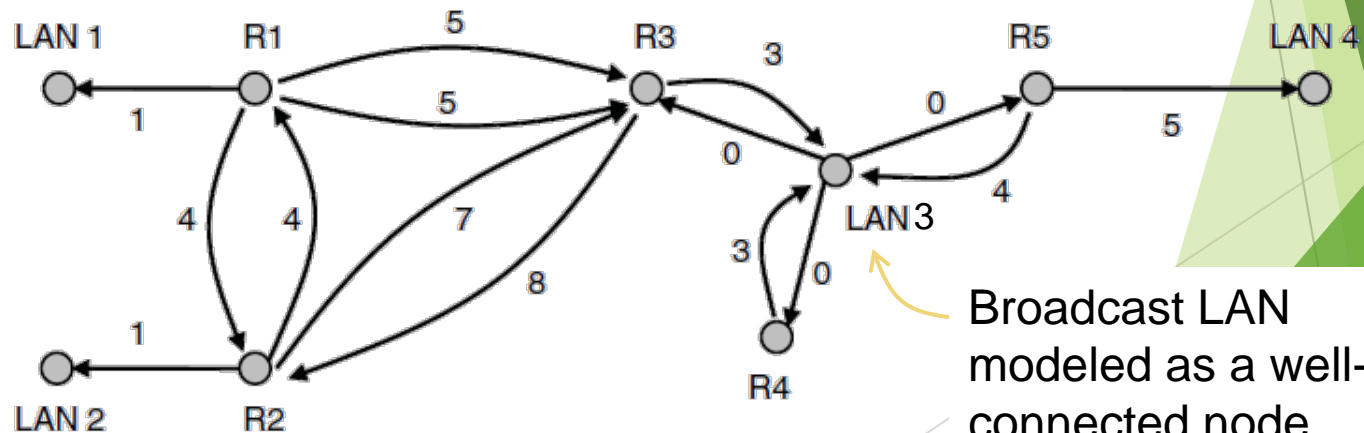
OSPF computes routes for a single network (e.g., ISP)

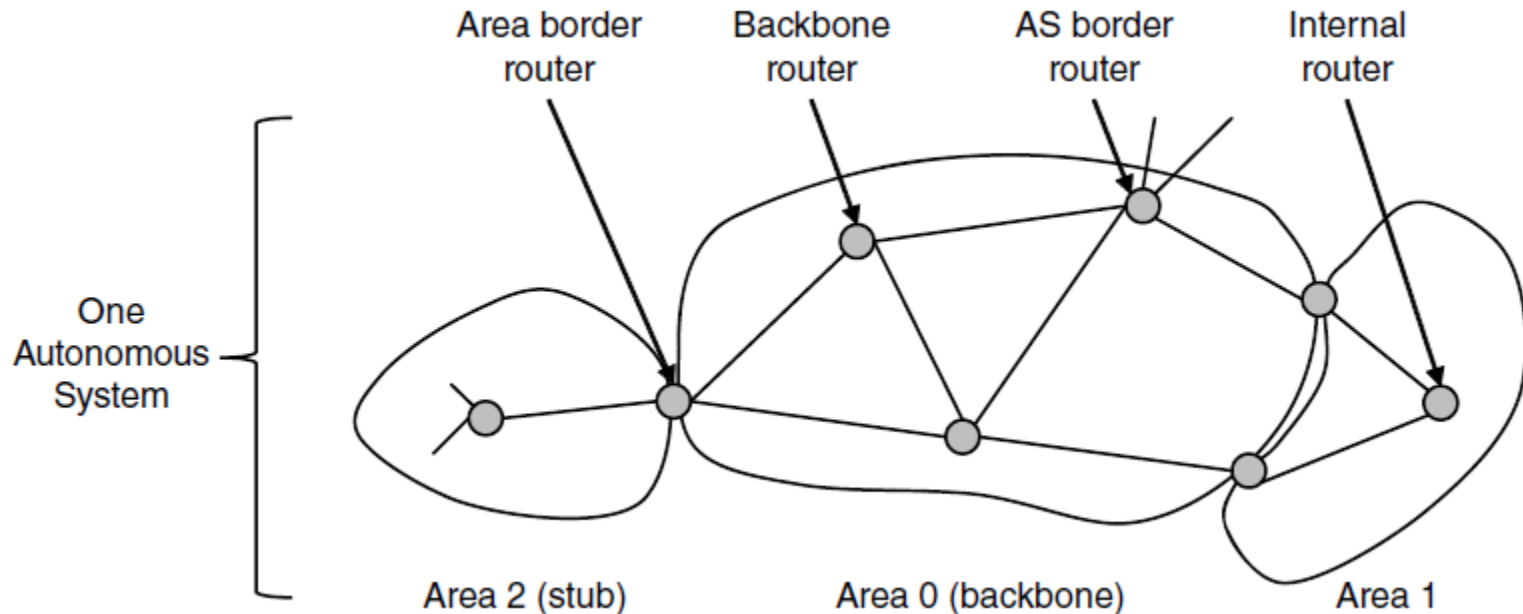▶ Models network as a graph of weighted edges

Network:

Graph:

Broadcast LAN modeled as a well-connected node

OSPF— Interior Routing Protocol (2)

OSPF divides one large network (Autonomous System) into areas connected to a backbone area

▶ Helps to scale; summaries go over area borders

OSPF— Interior Routing Protocol (3)

OSPF (Open Shortest Path First) is link-state routing:

- ▶ Uses messages below to reliably flood topology
- ▶ Then runs Dijkstra to compute routes

Message type	Description
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

BGP— Exterior Routing Protocol (1)

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

BGP— Exterior Routing Protocol (2)
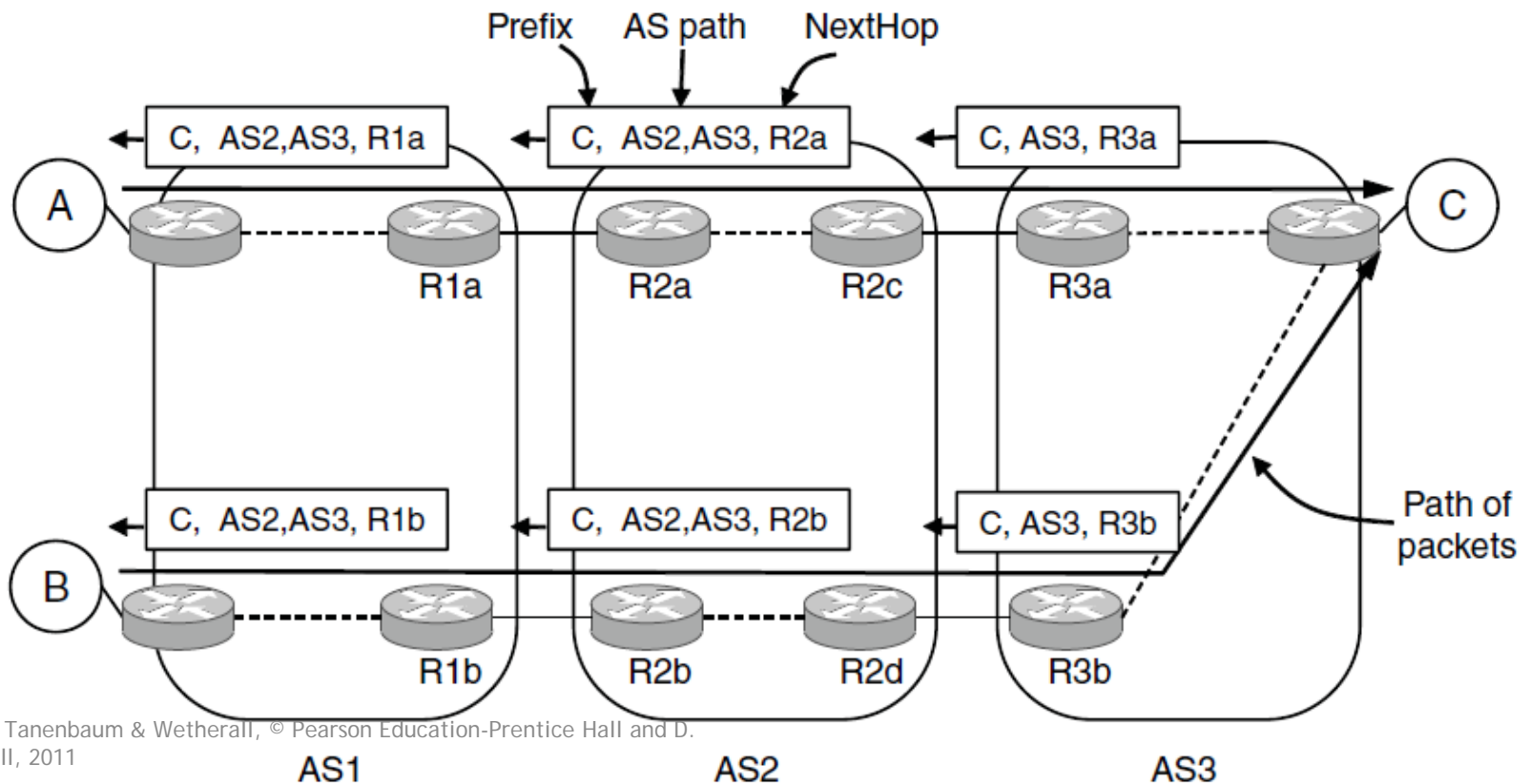
Common policy distinction is transit vs. peering:

▶ Transit carries traffic for pay; peers for mutual benefit

▶ AS1 carries AS2↔AS4 (Transit) but not AS3 (Peer)
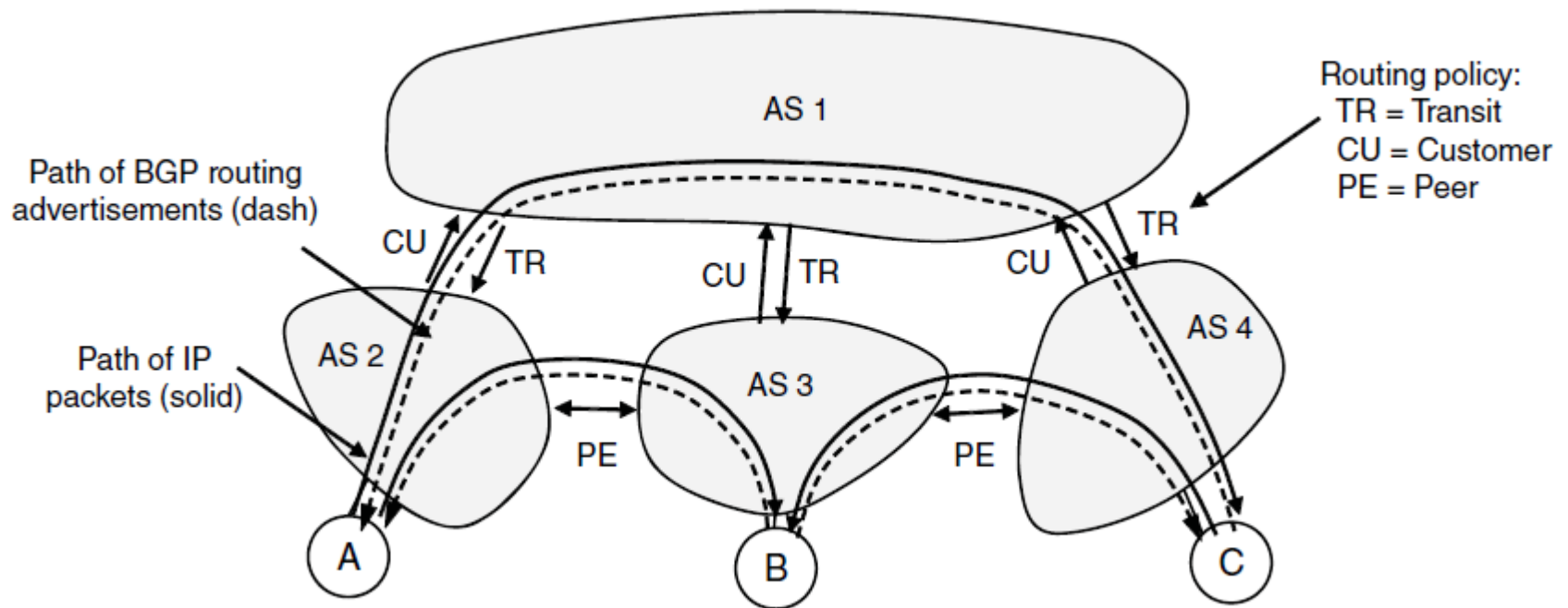
BGP— Exterior Routing Protocol (3)

▶ 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
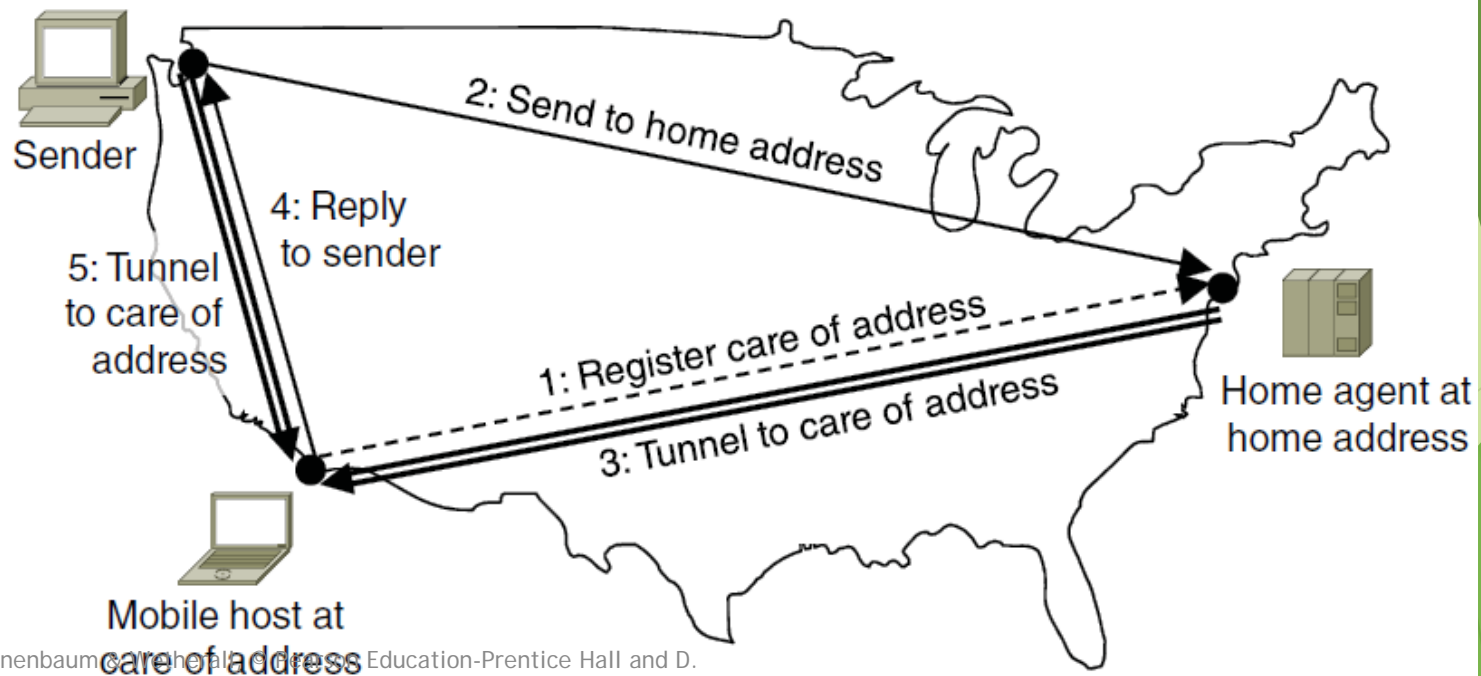
Routes computed by protocols such as PIM:

- ▶ 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

CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011

End

Chapter 5