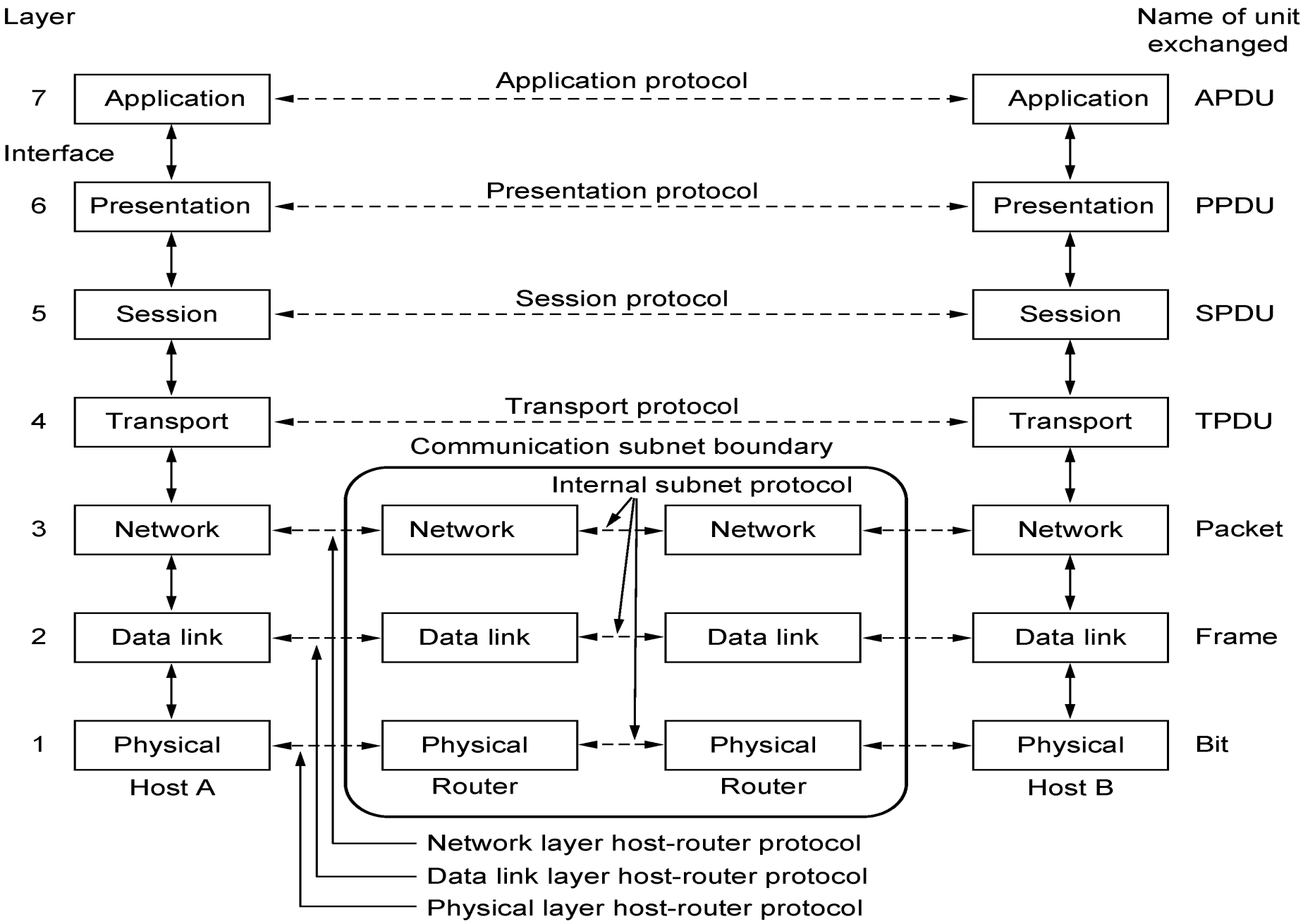
Network Layer

Chapter 5

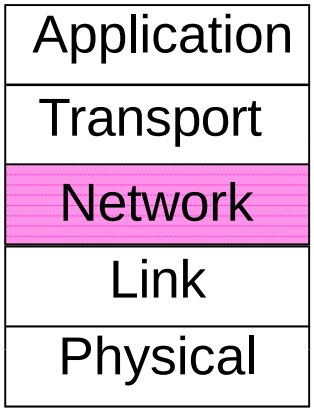
- Responsible for getting packets from the source to destination through INTERMEDIATE HOPS
- Data link layer: only between two points
- The network layer deals with end to end transmission
  - Should know the topology of the network
  - Should not overload some routes and underload other routes.
  - Should deal with different networks



The OSI reference model.

The Network Layer

Responsible for delivering packets between endpoints over multiple links



Network Layer

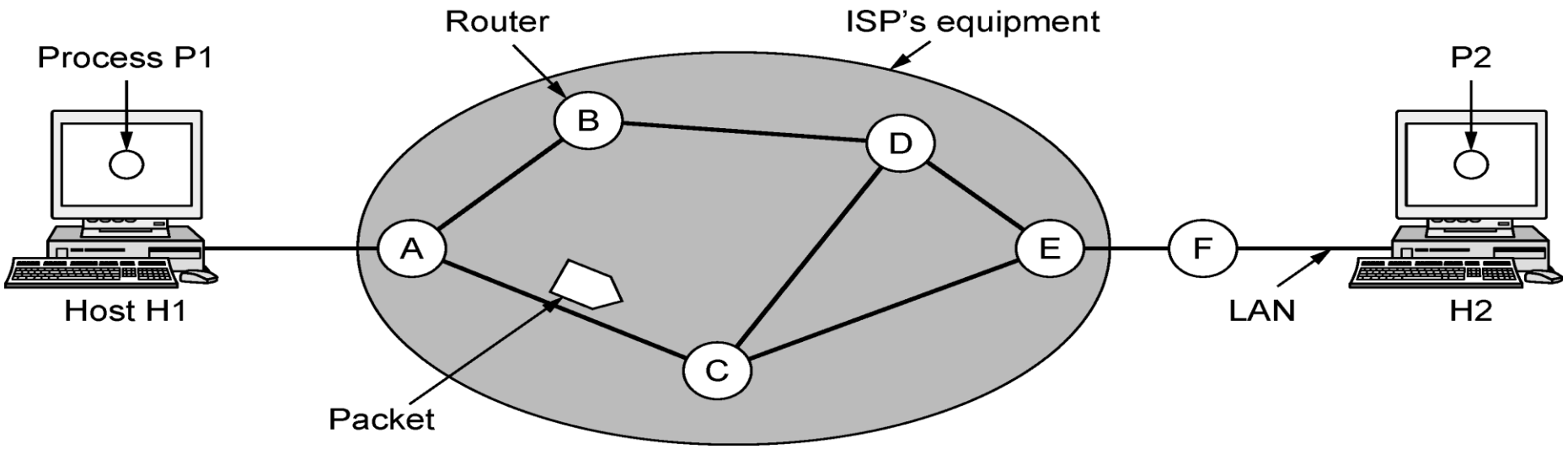
- Design Issues
- Routing Algorithms
- Congestion Control
- Quality of Service
- Internetworking
- Network Layer of the Internet

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

- Components of the network layer
  - ISP equipment (routers connected by transmission lines)
  - Host H1 is directly connected to one of the router
    - Home computer plugged into MODEM
    - H2 is a LAN, with an office Internet with a router, F, owned and operated by the customer.
    - The router has a leased line to the ISP's equipment.
    - F is not part of ISP, but it runs the same algorithms as the ISP's routers.
- Host sends the packet to the nearest router, which will be forwarded to the next router along the path until it reaches the destination.



The environment of the network layer protocols.

Services Provided to the Transport Layer

- Properties of services
  - The services should be independent of router technology
  - Transport layer should be shielded from the number, type, and topology of the routers present.
  - The network addresses made available to transport layer should use a uniform numbering plan across LANs and WANs.

- The freedom resulted into two thoughts
  - First: Network service should be connectionless with SEND PACKET and RECEIVE PACKET and little more.
    - No ordering and flow control should be done.
    - Each packet should carry a destination address.
  - Second: network should be reliable and connection-oriented service based on the experience of telephone service.
    - Without connections, quality of service is difficult to achieve especially for real-time traffic.

- First, connectionless service has dominated.
- Later, connection-orient service dominated
- Currently, both are widely used.

Connectionless service: datagrams

- Packets are injected as datagrams (analogy is telegram).
- No advance setup is needed
- Network is called datagram network.
  - H1 handover long message to transport layer which instructs to deliver packet to P2 on H2. The transport layer runs in H1 (in the operating system). It prepends transport header and handover it to network layer (another procedure)

- Network layer breaks it into packets and sends each of them to router A using point-to-point protocol, PPP.
- ISP decided where to send packets. It has to take routing decisions.

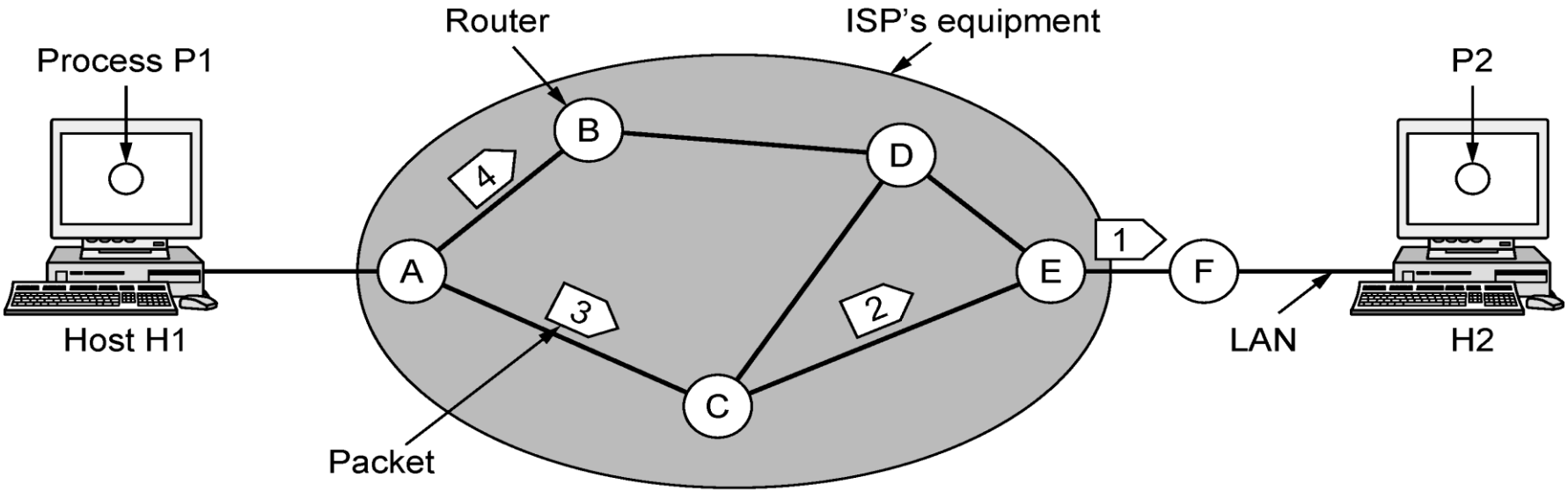
- The algorithm which does routing decisions are called **routing algorithm**.

- In this chapter, we study routing algorithms.
- Internet protocol (IP) is the basis of the Internet, individually forwards each packet. 32 bits in IPv4 and 128 bits in IPv6.

Connectionless Service – Datagrams

Packet is forwarded using destination address inside it

- Different packets may take different paths



A's table (initially)

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

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

Routing within a datagram network.

Connection-oriented: Virtual Circuits

- Idea: Avoid having a new route for every packet.
- Establish a connection. When a connection is established, the connection set-up is stored in tables.
  - The route is used for all traffic flowing over the connection.

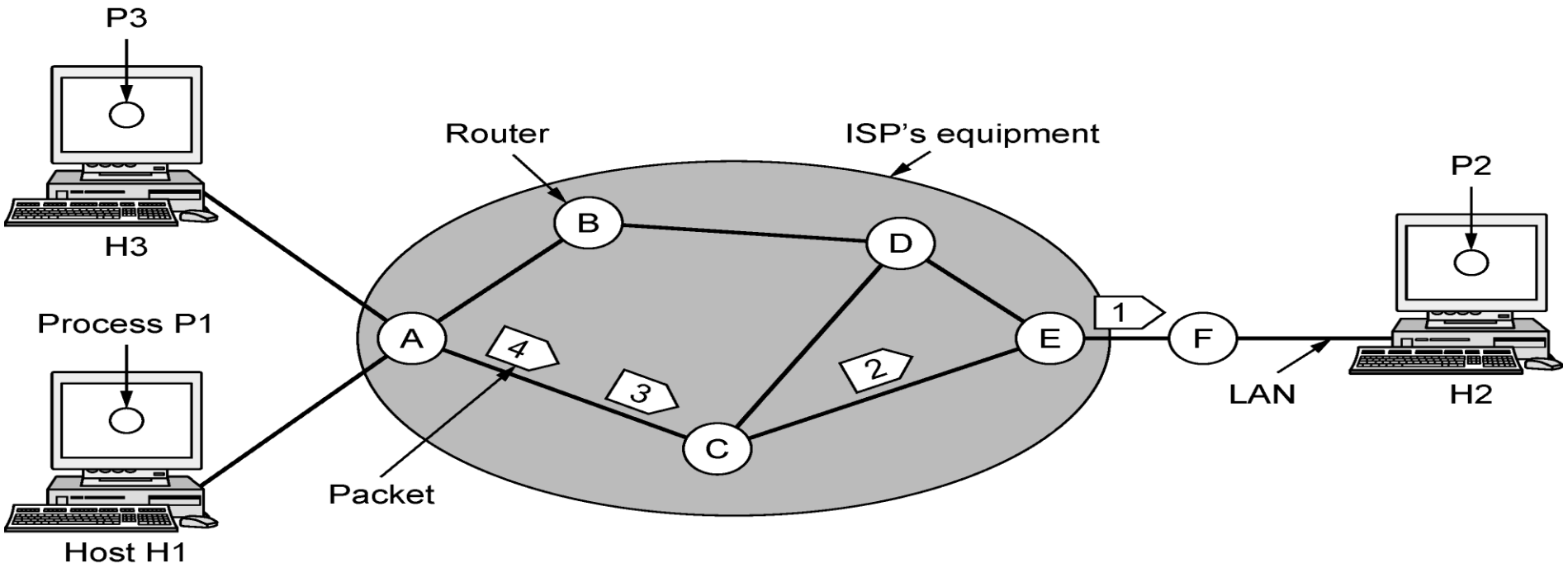
- H1 established connection 1 with host H2. The first line of A's table says that if a pack arrives with connection 1 it should be forwarded to C and given connection id as 1.
- If H3 also wants to establish a connection to H2, it choses a connection identifier 1 and tells the network to establish a connection.

- There is a conflict as A has already used id 1. So, A assigns a different identifier. This process is called label switching. An example of the protocol is MPLS (MultiProtocol label Switching). It wraps packets with 20-bit connection identifier or label.

Connection-Oriented – Virtual Circuits

Packet is forwarded along a virtual circuit using tag inside it

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



A's table

H1	1
H3	1

In

C	1
C	2

Out

C's table

A	1
A	2

E	1
E	2

E's table

C	1
C	2

F	1
F	2

Routing within a virtual-circuit network.

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

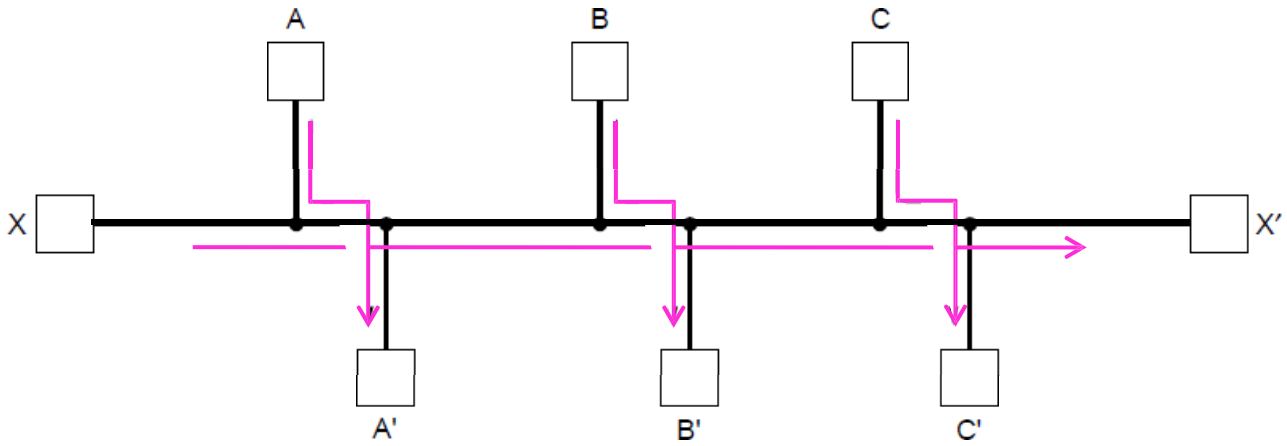
- The decision has to be made for every packet as best route may be changed over last time.
- Network layer design
  - Algorithms and data structures
- Routing and forwarding
  - Forwarding: a process which takes each packet and forwards it to outgoing line.
  - Routing: A process which updates the routing tables.
    - Routing algorithms play a major role.

- Properties of routing algorithm
  - Correctness, simplicity:
  - Robustness: it should run for years; hosts fail, routers crash, software fails
  - Stability: Routng algorithm should converge and reach equilibrium.
  - fairness, efficiency: Difficult to achieve: For example, in case of high traffic between A and A', B and B', C and C', the traffic between X and X' will be shut off. So compromise is required.

Routing Algorithms (2)

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

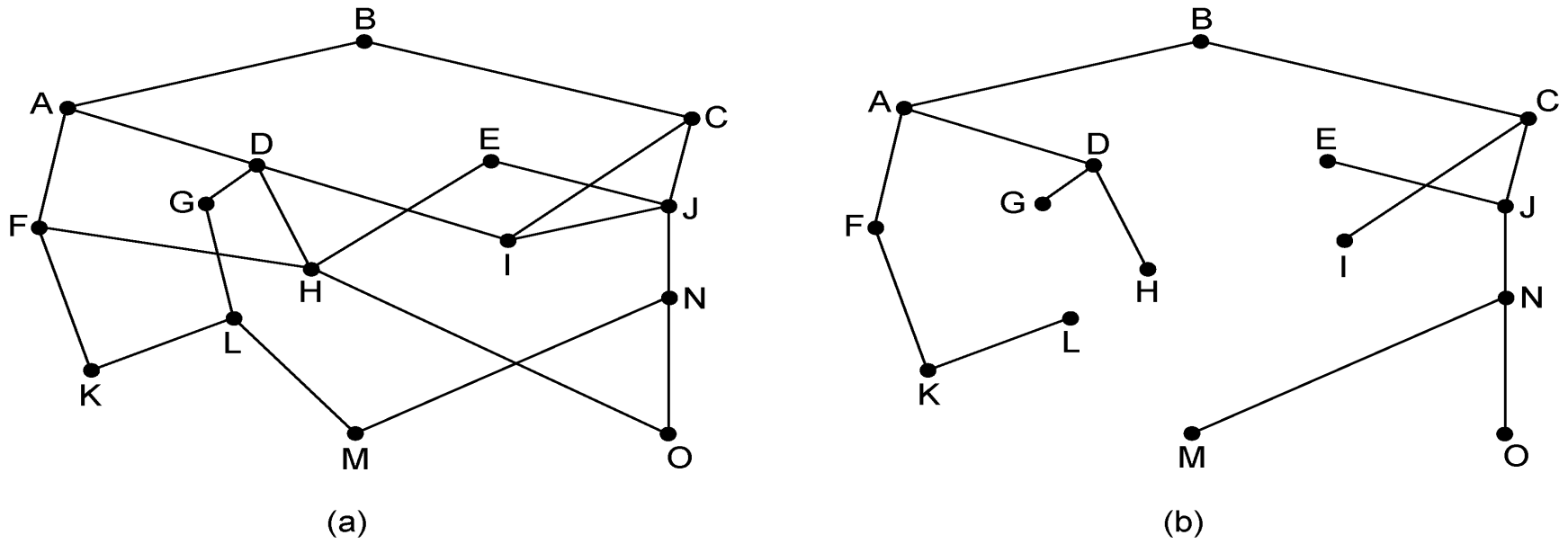
Routing Algorithms

- Nonadaptive algorithms
  - Decisions do not depend on the state or topology
  - When the network is booted, routing tables are fixed.
  - Also called static routing algorithms

- Adaptive algorithms
  - Decisions are based on on the state or topology
  - Dynamic routing algorithms

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
- Set of all optimal routes from all sources to a given destination form a **sink tree**
- Goal of routing algorithm: discover and use sink trees for all the routers.
- The optimality principle acts as a Benchmark for comparison



(a) A network. (b) A sink tree for router B.

Shortest Path Algorithm (1)

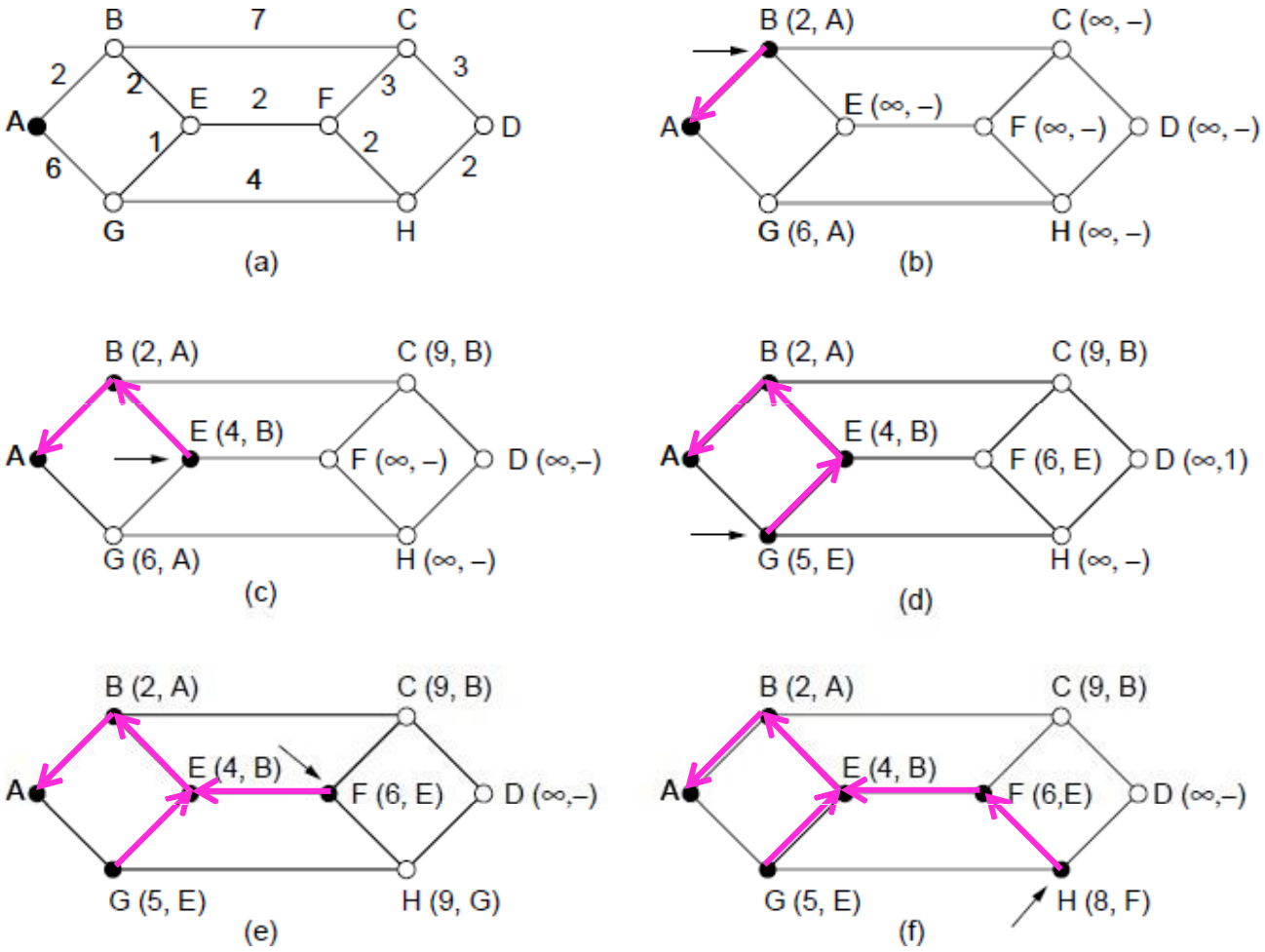
Dijkstra's algorithm computes a sink tree on the graph:

- 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
- Generates vast numbers of duplicate packets.
- Method 1:
  - Flooding with an hop count: Put hop counter and decrement hop counter for each node.
  - -ve: generates exponential number of duplicate pockets
- Method 2
  - Track the packets. Stop, if it is already flooded, i.e., avoid sending second time.
    - Put a sequence number for each source in the packet. Maintain the list at each node.

Flooding

- Flooding is not practical, but has several advantages
  - It ensures that packet reaches every node.
  - Flooding is tremendously robust.
    - Even if large number of routers are damaged, the packet reaches the destination.
  - Flooding produces shorter delay

- Flooding can be used to compare other algorithms.

Distance Vector Routing (1)

- Dynamic
- Distance vector is a distributed routing algorithm
  - Shortest path computation is split across nodes
  - Each router maintains the table giving the best known distance to each destination

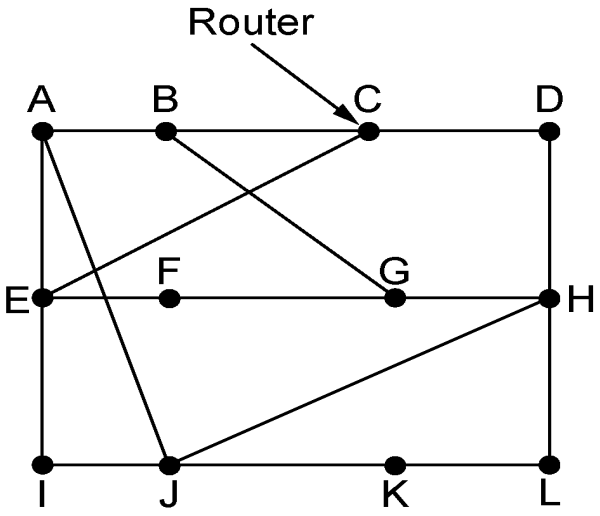
Bellman-Ford Algorithm (followed in ARPANET):

- Each node knows the 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 (1)

- Once every T sec, each router sends to each neighbor a list of estimated delays to the destination. It also receives similar list from its neighbors and calculated the new routing table.
- Example
  - Four columns show the delay vectors received from neighbors. A claims to have a 12 msec delay to B, 25msec delay to C, and 40 sec delay to D and so on.
  - Suppose, J has measured or estimated its delay to its neighbors A, I,H, and K as 8, 10, 12, and 6 respectively.
  - J computes time delay to G as follows.
    - One route: J,A,..G = 8 (JA) +18 (AG)=26
    - Second route: J I ..G=41(31+10)
    - Third route: J H..G= 18(6+12)
    - Fourth route: JK..G=37 (31+6)
    - The best route is 18.

Distance Vector Routing (2)



(a)

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

Vectors received from J's four neighbors

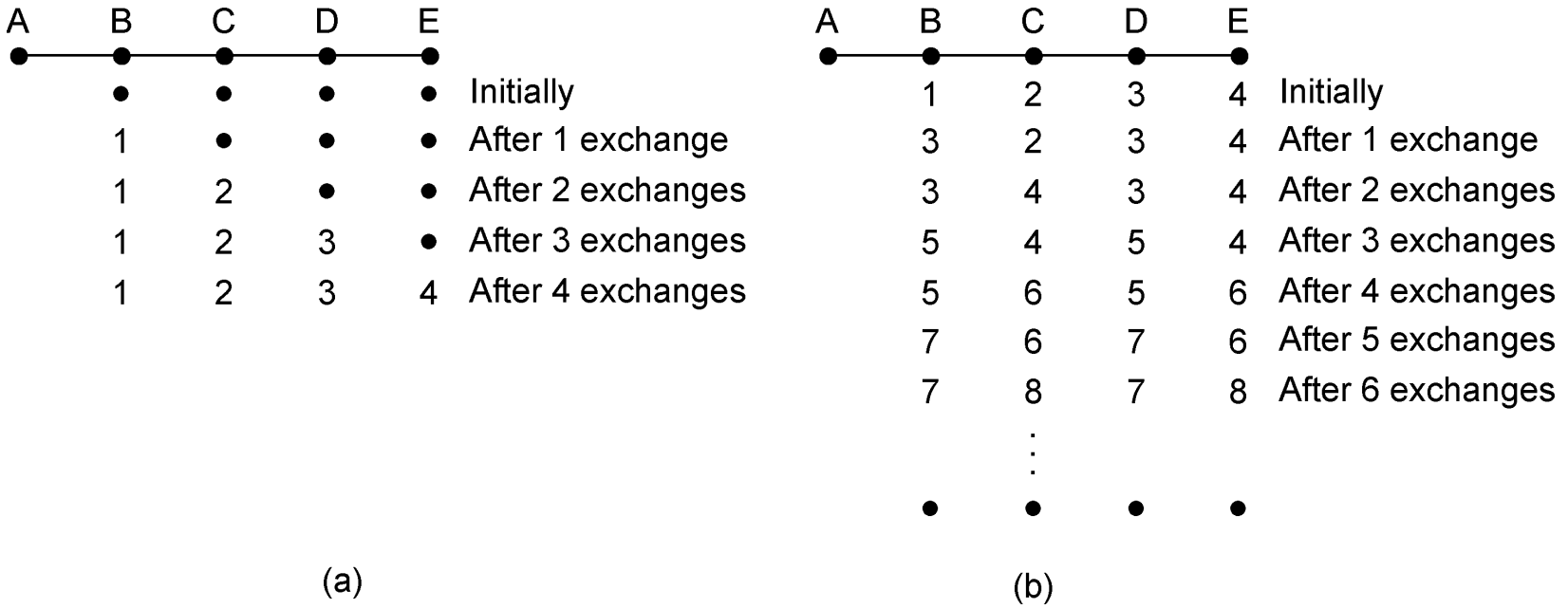
New routing table for J

(b)

(a)A network. (b)Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

The Count-to-Infinity Problem

- Convergence: Though it converges to correct answer, it converges slowly.
- Responds rapidly to good news, but very slowly to bad news.
- In Figure (a) shows how good news spreads quickly.
- Figure (b) demonstrates count to infinity problem.



The count-to-infinity problem.

Link State Routing (1)

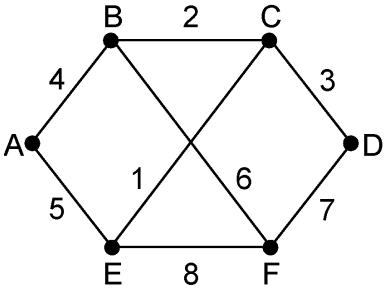
- Distance vector routing is replaced by link state routing.
- More computation but simpler dynamics
- Widely used in the Internet (OSPF, ISIS)

Algorithm: Each router

1. Discovers its neighbors and learn their network addresses.
   - After booting, node sends HELLO packet to each point-to-point line.
2. Set the distance or cost metric to each of its neighbors
3. Construct a packet with the learning.
4. Send this packet and receive packet from all other routers.
5. Compute the shortest path to every other router

Link State Routing (2) – LSPs

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



(a)

Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

(b)

(a) A network. (b) The link state packets for this network.

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

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

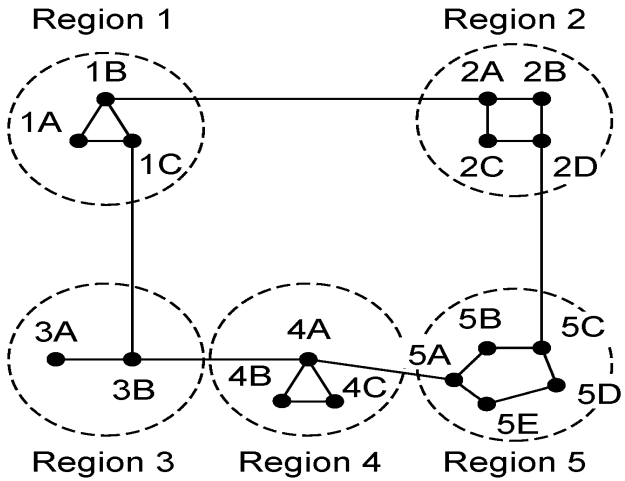
The packet buffer for router B in Fig. 5-12(a).

Hierarchical Routing

- As network grows, the router tables grow proportionately.
  - More memory and CPU time
  - More bandwidth
- After certain limit, it is not possible to send routing table to every other router.
- Hierarchical routing
  - Routers are divided into regions.
  - Each router knows how to route packets to other routers within the region.
  - Hierarchy may consists of multiple levels.

Hierarchical Routing

Hierarchical routing reduces the work of route computation but may result in slightly longer paths than flat routing



(a)

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

(b)

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

(c)

Hierarchical routing.

Broadcast routing

- Broadcast sends a packet to all nodes
- Simple algorithm: send the packet to each destination.
  - It is slow and wastes bandwidth
  - Source should have a list of all destinations.
- Multi-destination routing
  - Each packet contains a list of destinations or a bitmap indicating desired destinations.
  - The router generated new copy of the packet for each output line and includes in the packet only those destinations which are on the line.
  - After sufficient number of hops the packet will have only one destination
  - Bandwidth is used efficiently.

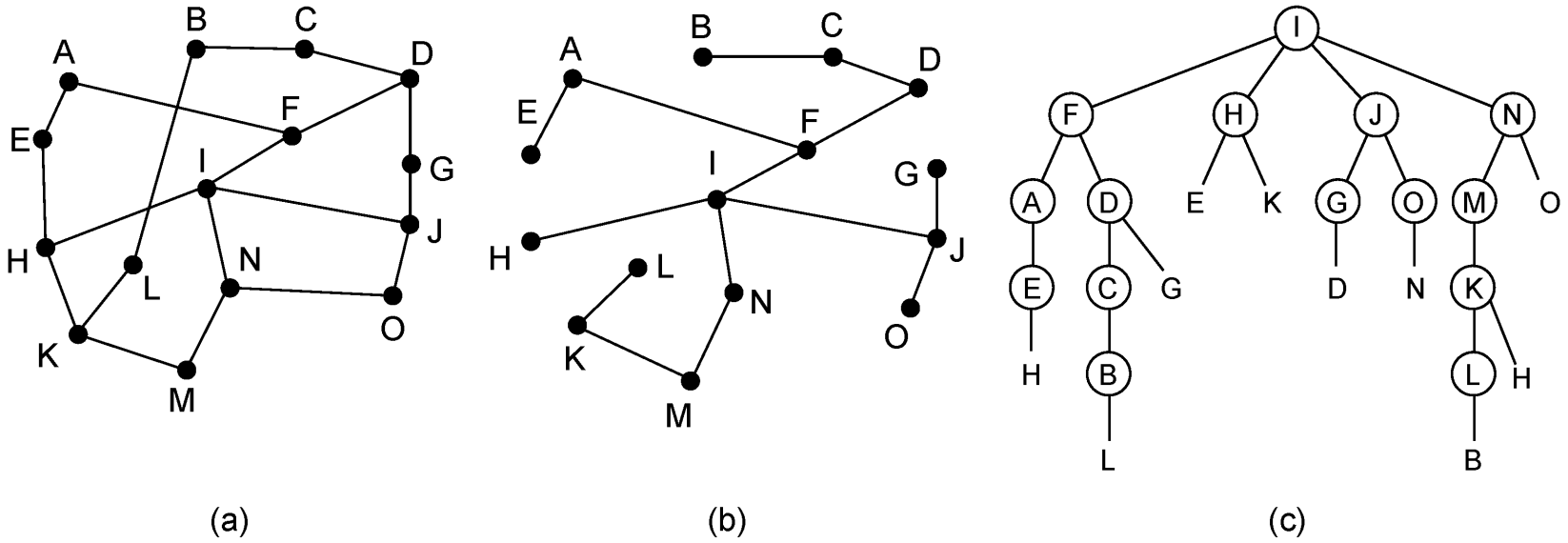
Broadcast routing

- Flooding is another broad cast technique
- Reverse path forwarding (figure (c))
- When a broadcast packet arrives to the router, the router checks to see if the packet arrived on the link that is normally used for sending packets toward the source of the broadcast.
  - If so, there is a chance that broadcast packet might have followed the excellent route.
  - The router forward the packet to all links except incoming links.
  - If the packet arrives from other link than the preferred one, the pack is discarded as a likely duplicate.
  - After 5 hops, with 24 packets broadcasting terminates.
  - +ve: easy to implement and efficient; no sequence numbers are required.
- Spanning tree algorithm: figure (b)
  - Minimum of 14 packets
  - -ve: each router should have knowledge of spanning tree

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



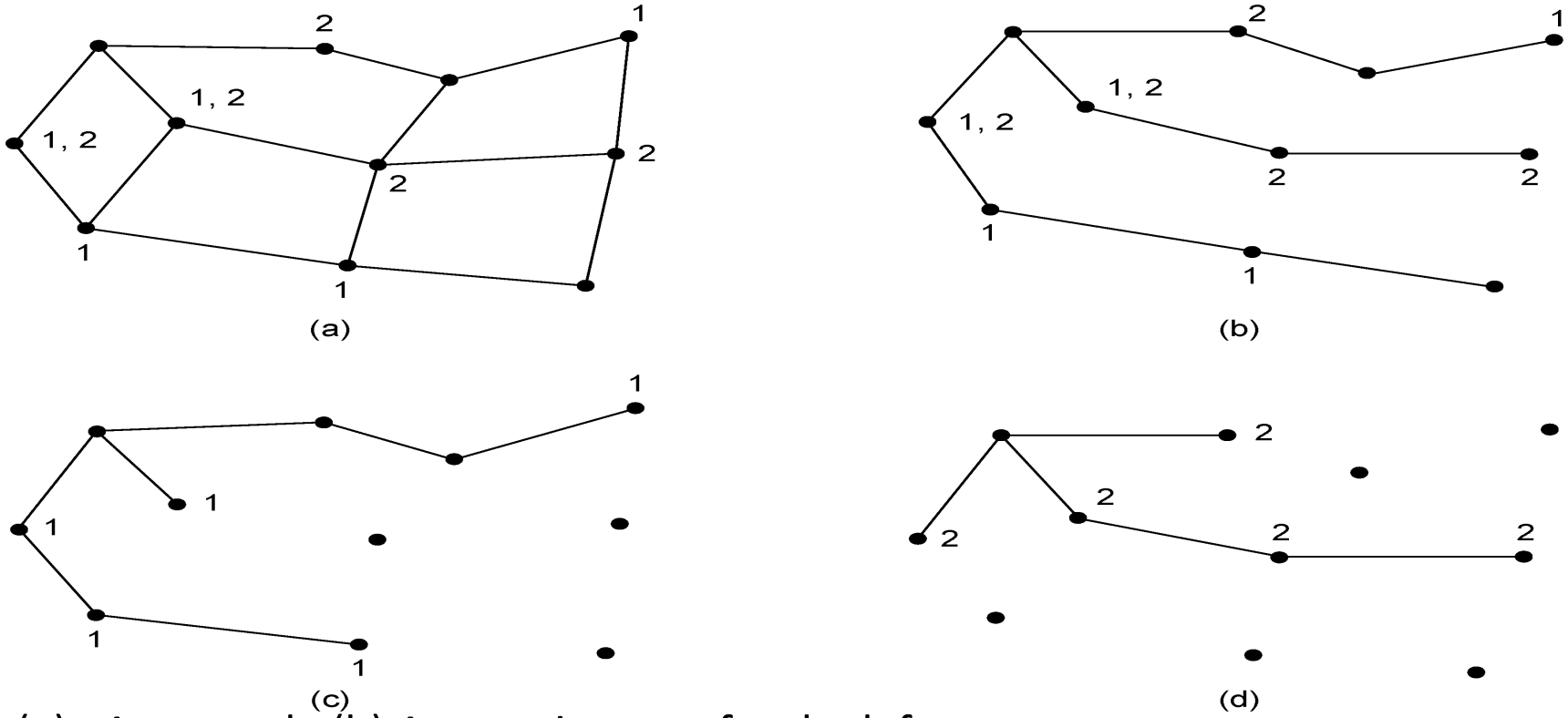
Reverse path forwarding. (a) A network. (b) Sink tree for router *I*. (c) The tree built by reverse path forwarding from *I*.

Multicast Routing (1) – Dense Case

- Send a message to well-defined group, numerically large by small than the size of the network
  - Example: live video
- Multicast sends to a subset of the nodes called a group
- Routing algorithm: multicast routing
- All multicasting schemes require a way to create and destroy groups and identify which routers are the members of the group.

Multicast Routing (1) – Dense Case

- If the group is dense, employ broadcasting
  - Remove those nodes which are not the members in the spanning tree.
  - Multicasting spanning tree
    - Use link state routing to identify the nodes.

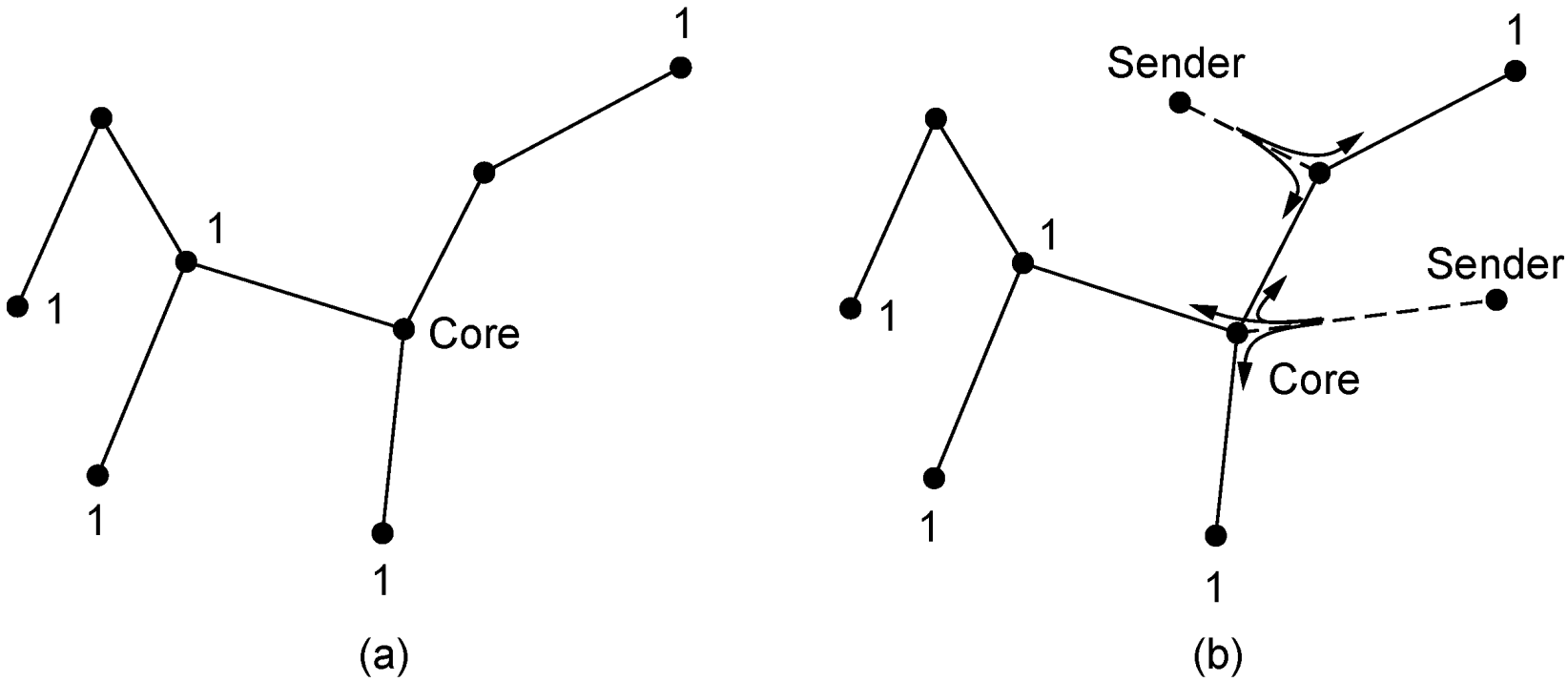


(a) A network. (b) A spanning tree for the leftmost router.
(c) A multicast tree for group 1. (d) A multicast tree for group 2.

Multicast Routing (2) – Sparse Case

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

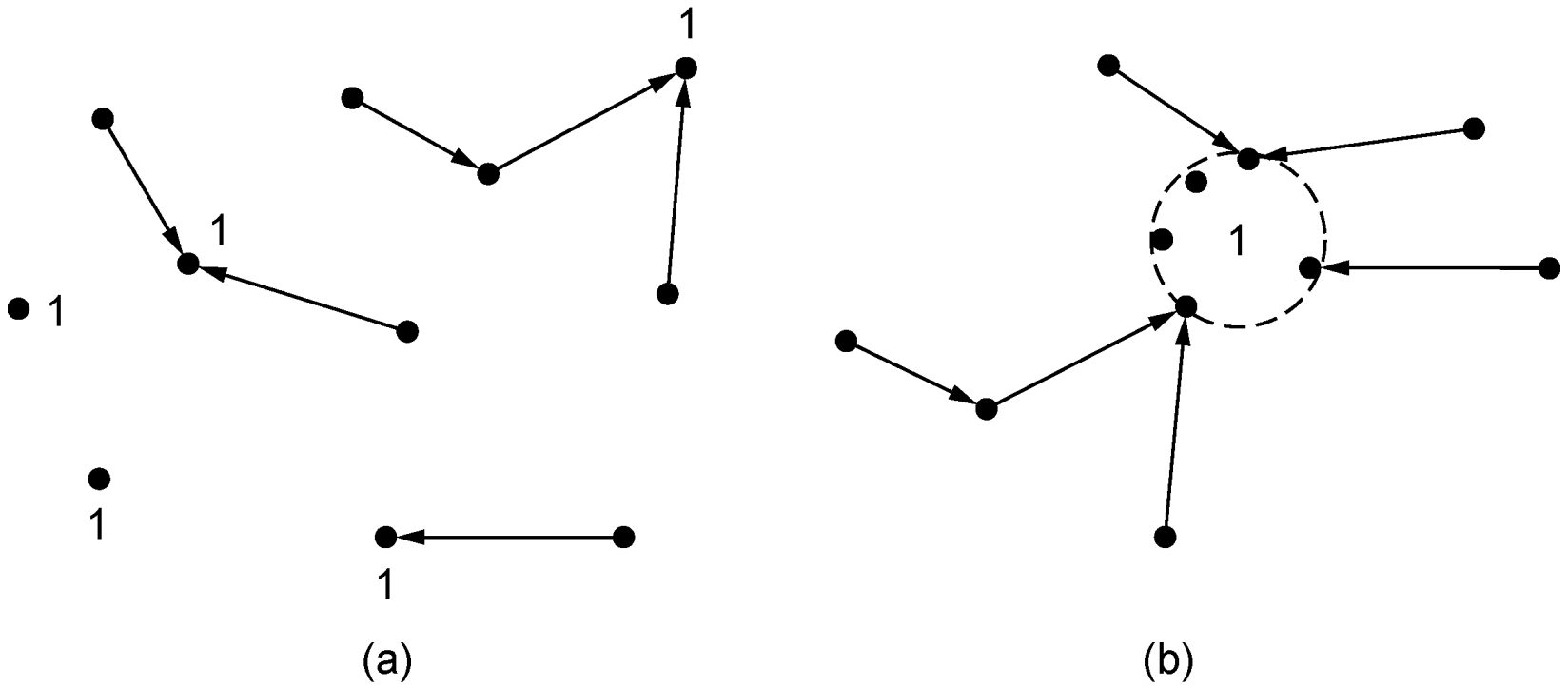
- All the members agree on a tree.
- Tree is the sink tree from core node to group members
- Multicast heads to the core until it reaches the CBT



(a) Core-based tree for group 1. (b) Sending to group 1.

Anycast Routing

- Anycast sends a packet to one (nearest) group member
  - Used for time of the day, DNS service
  - Falls out of regular routing with a node in many places

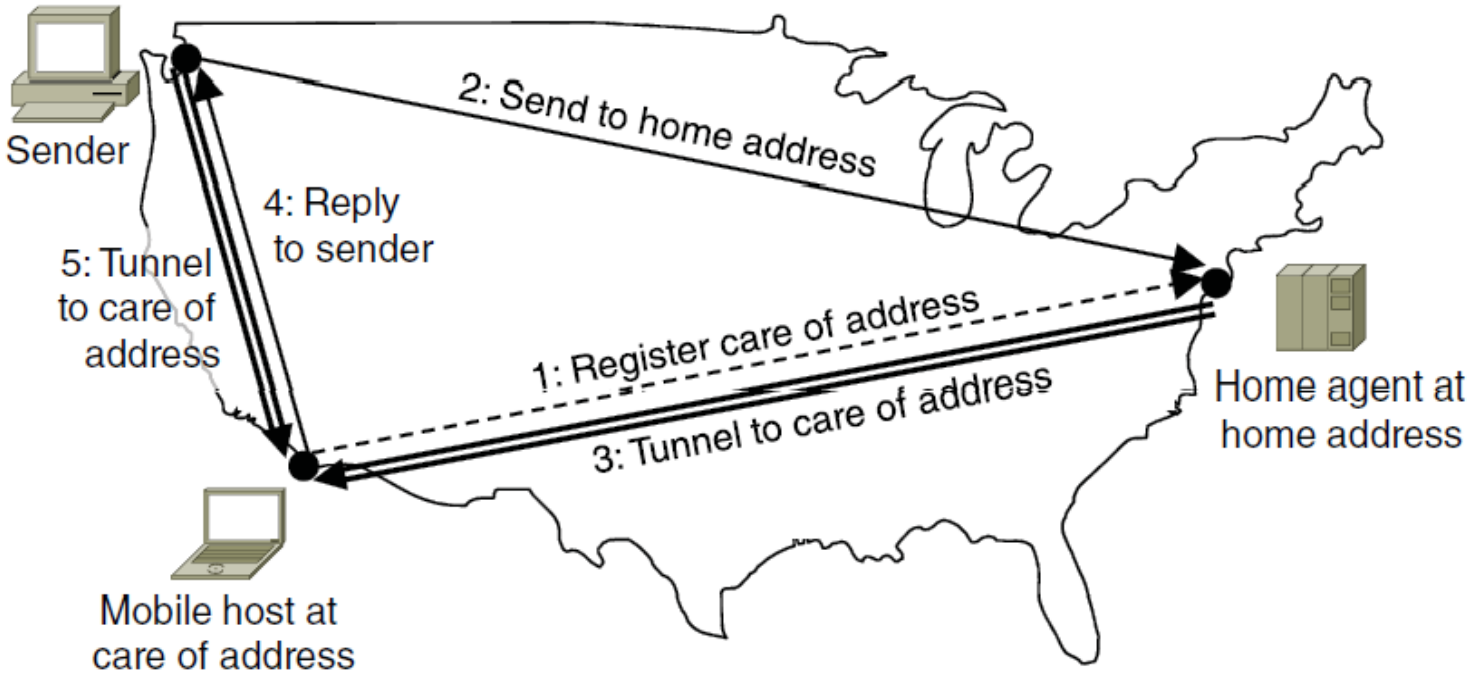


(a) Anycast routes to group 1. (b) Topology seen by the routing protocol.

Routing for Mobile Hosts

Mobile hosts can be reached via a home agent

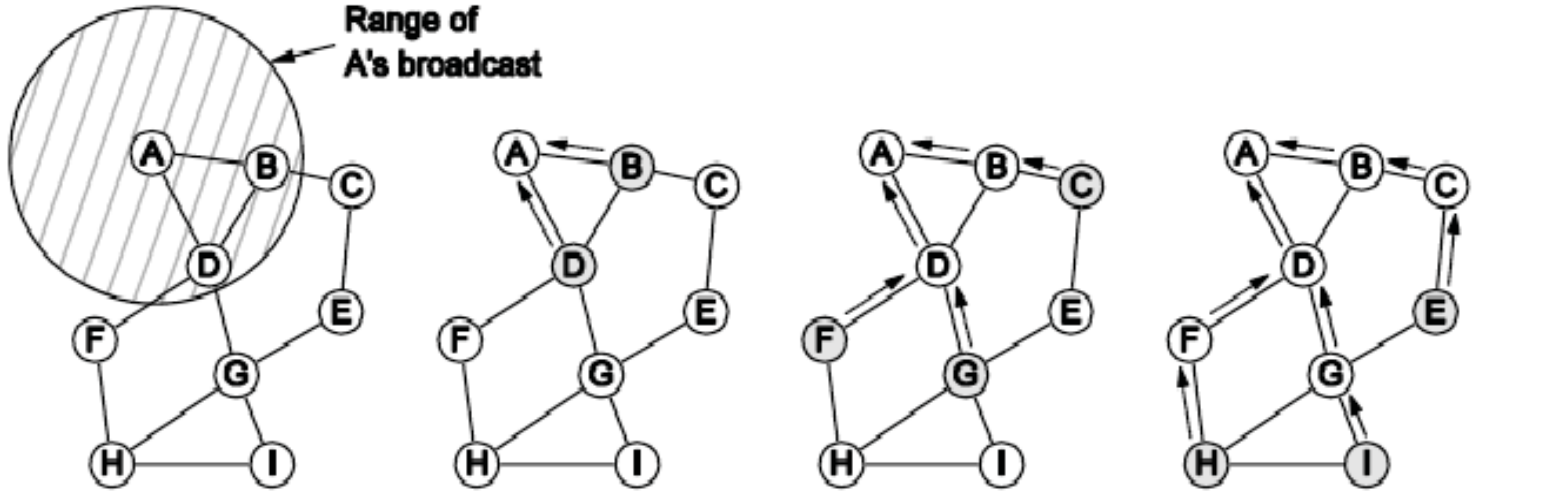
- Fixed home agent tunnels packets to reach the mobile host; reply can optimize path for subsequent packets
- No changes to routers or fixed hosts



Routing in Ad Hoc Networks

The network topology changes as wireless nodes move

- Routes are often made on demand.



A's starts to find route to *I*

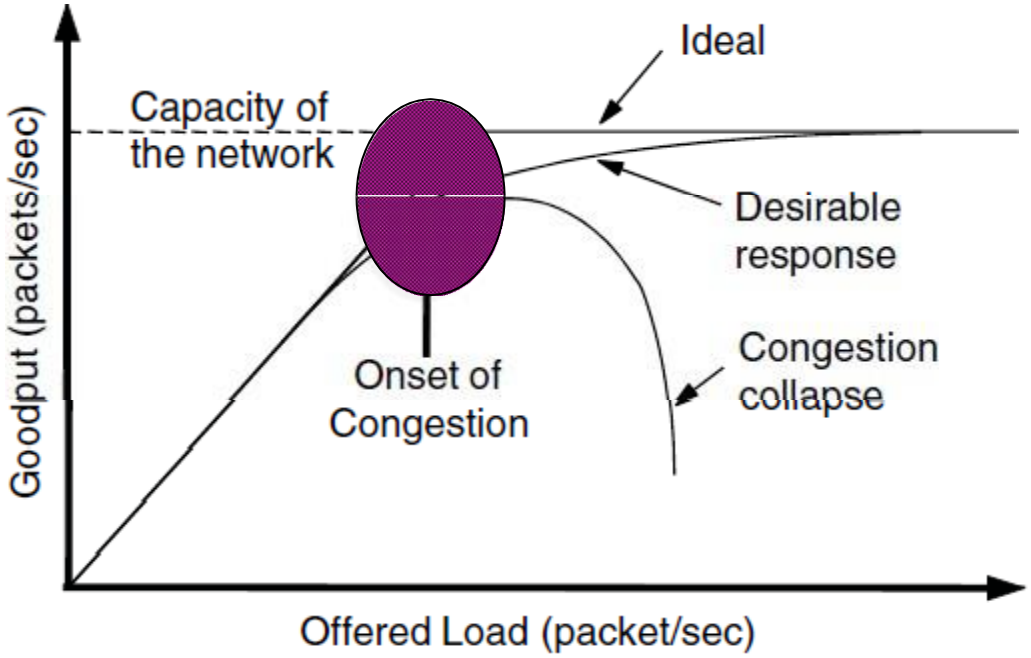
A's broadcast reaches *B* & *D*

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

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

Congestion Control (1)

- Congestion is a situation: Too many packets present in the network causes network delay and loss
- Both network and transport layers share the responsibility to control congestion.
- Effective way is to reduce the load at the transport layer.
- Goodput (=useful packets) trails offered load



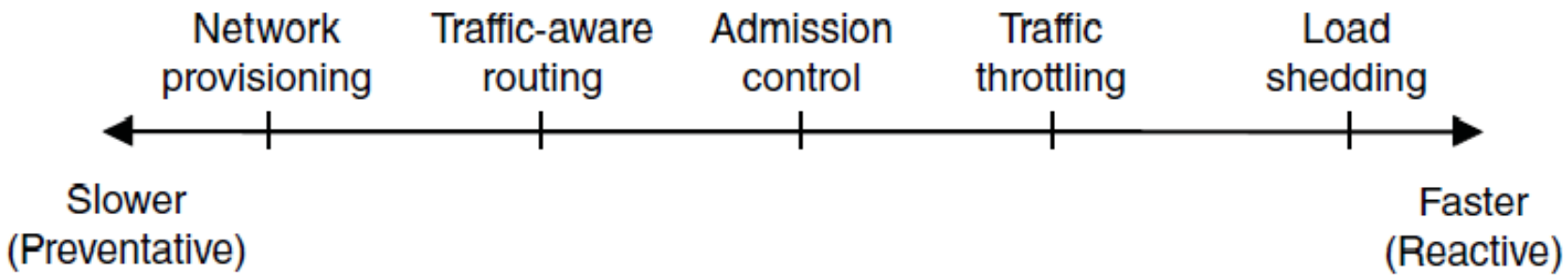
Congestion Control (2)

- Network experiences a congestion collapse
- If routers will have infinite memory, congestion gets worse
  - Packets gets timed out.
- Difference between congestion control and flow control
- Congestion control is related to whole network. Involves the behavior of all the hosts and routers.
- Flow control relates to the traffic between a particular sender and a particular receiver.
- Best method is to get host a “slow down”.

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)



Congestion Control (1)

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

- We look at the Network portion here

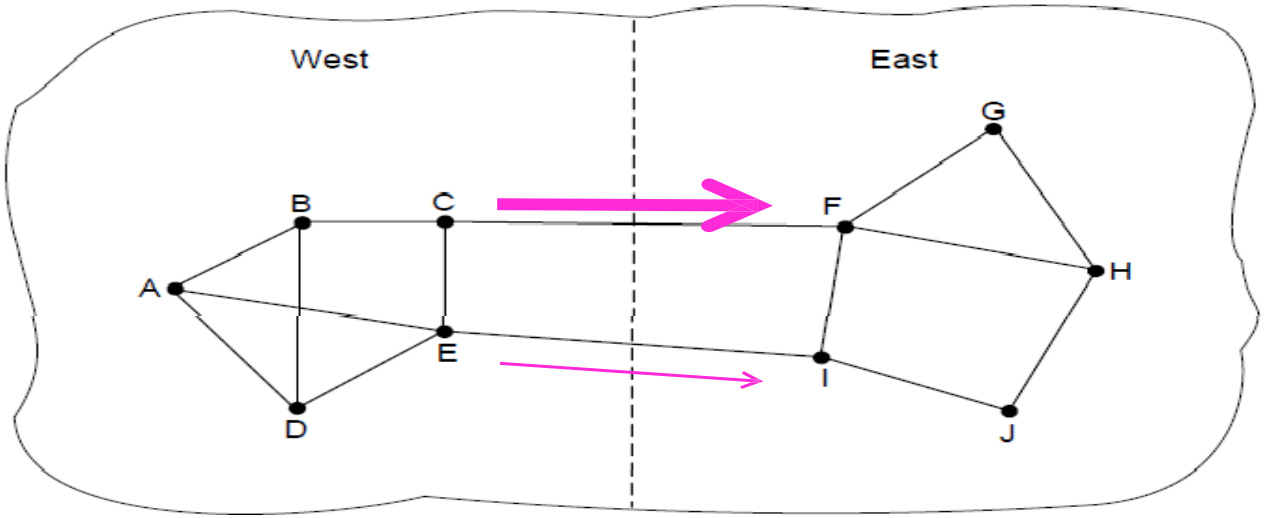
- Traffic-aware routing »
- Admission control »
- Traffic throttling »
- Load shedding »

Network provisioning

- Links and routers that are heavily utilized are upgraded at the earliest opportunity.
  - By considering short-term and long-term traffic trends.

Traffic-Aware Routing

- Choose routes depending on traffic, not just topology
- Link weight is a function of the link bandwidth, propagation delay, measured load (average queuing delay)
- Routing tables are updated accordingly.
- E.g., use *EI* for West-to-East traffic if *CF* is loaded
- -ve: routing tables may oscillate wildly leading to erratic routing and other problems

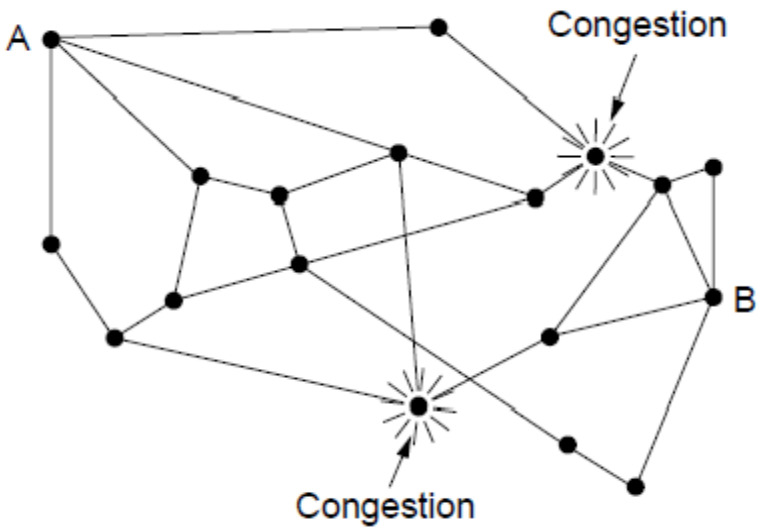


Admission Control

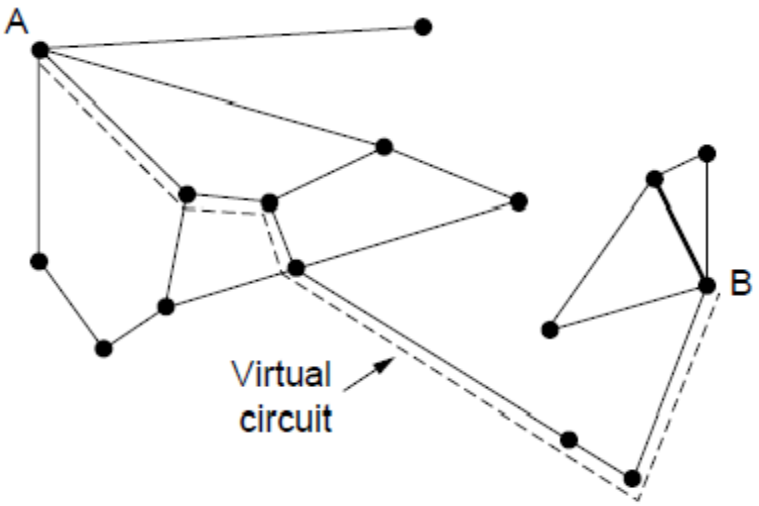
- Idea: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.
  - Attempt to set up a virtual circuit may fail.
  - Example: telephone system
- Issue is handling **bursty traffic**
  - Video is easy to handle
  - Traffic due to web browsing is very bursty
- Admission control can be combined with traffic ware routing
- Redraw the network topology based on information about congested routers.
  - For example, normally connection between router A and B passes through one of the congested routers. Based on the information, we can redraw the network.

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



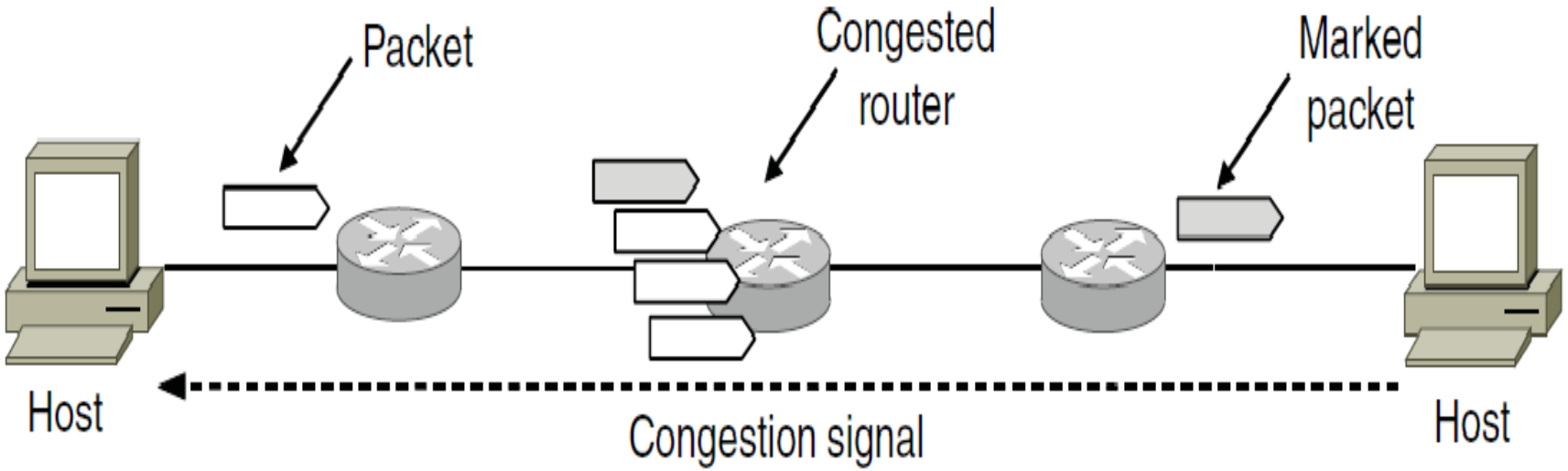
Uncongested portion and route *AB* around congestion

Traffic Throttling

- When the congestion is imminent, network should tell the senders to throttle back their transmissions and slow down.
- Congestion avoidance
- Congested routers signal hosts to slow down traffic for Datagram networks and virtual circuit networks.
- Two problems have to be solved for each approach
- First: router must determine when congestion is approaching, before it has arrived.
  - Queuing delay captures congestion experienced by packets
- Second: routers must send timely feedback to the senders that are causing congestion.
  - ECN (Explicit Congestion Notification) marks packets and receiver returns signal to sender

Traffic Throttling: feedback mechanisms

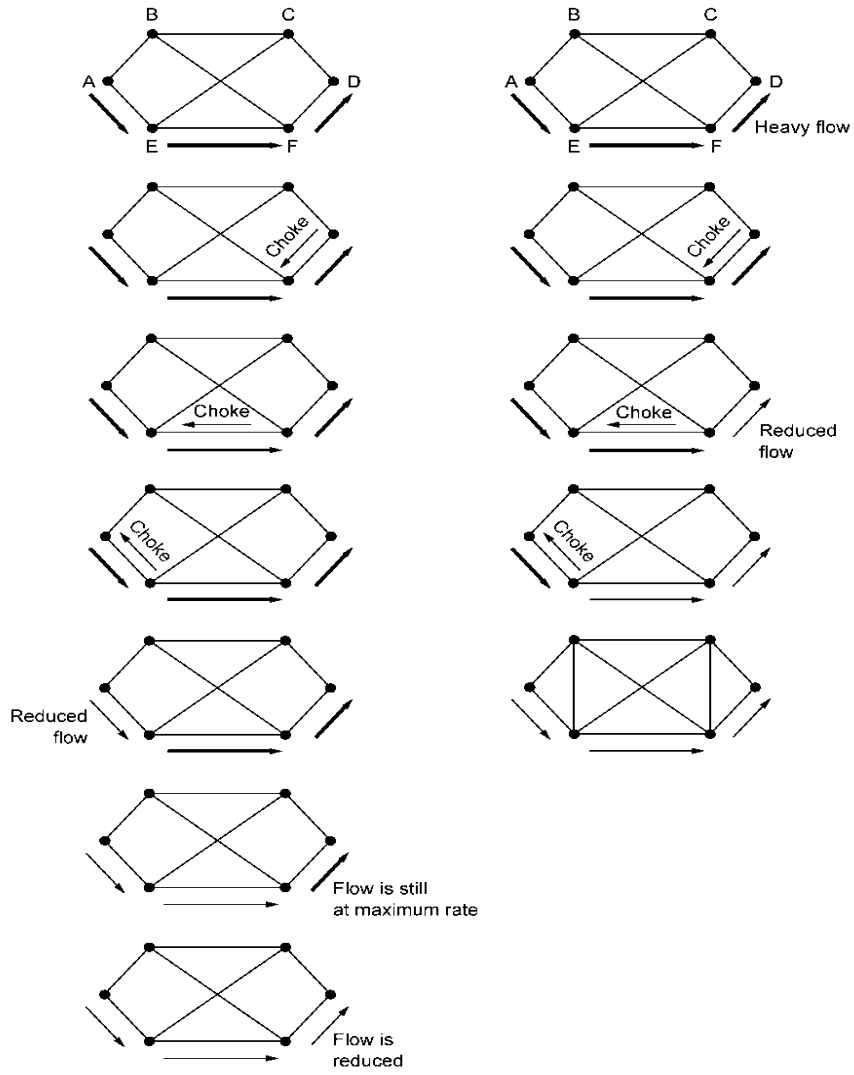
- Choke packets
  - Router selects the congested packet and sends the choked packet back to source host.
  - When a source gets the choked packet, it is supposed to reduced the 50% of traffic.
- Explicit congestion notification (ECN)
  - The router can tag any packet it forwards to signal that it is experiencing a congestion.
  - When a network delivers a packet, the received can inform the sender about congestion through ack. message. The design is called Explicit Congestion Notification (ECN).



Traffic Throttling: feedback mechanisms

- With ECN method, many new packets may be ttransmittted after the congestion.

- Alternative approach: Hop-by-Hop Backpressure
  - Have the choke packet take effect at every hop it passes through
  - Whenever a packet reaches F, F is required to reduce the flow to D.
- Example:
  - As soon as choke packet reaches F, F is required to reduce the flow to D.
    - It gives the D immediate relief
  - Similarly when a choke packet reaches E, E has to reduce the flow to F.



(a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

Load Shedding (1)

- When all else fails, network will drop packets (shed load)
- When routers are being inundated by packets that they can not handle, they just throw them away.
- Key question
  - Which packets to drop?
  - File transfer: old packet is important
    - WINE policy
  - Realtime traffic: new packet is important
    - MILK policy
- Application must mark the importance on the packets.
- **Random Early Detection**
  - Routers maintain running average of queue length
  - When a queue length exceeds, router starts dropping the packets at random.
  - Send will notice the loss as there are no ack. And slow down
  - Better performance.

Load Shedding (1)

- **Random Early Detection**
  - Routers maintain running average of queue length
  - When an average queue length exceeds, router starts dropping the packets at random.
    - Fast senders will have more packet drop
  - Sender will notice the loss as there are no ack. and slow down
  - Better performance.

Quality of Service

- So far, we have made effort to improve network performance
- However, some applications require stronger performance guarantees.
  - Example: Multimedia applications:
- We will focus of providing network performance with a sharp focus on providing quality of service.
- One solution is overprovisioning
  - More money is required
- With quality of service mechanisms, the network can honor performance guarantees at the cost of turning down some requests.

- Issues
  - What applications need from the network?
  - How to regulate the traffic that enters the network?
  - How to reserve the resources at routers to guarantee performance?
  - Whether the network can safely accept more traffic
- No single technique deals efficiently all these issues..

Quality of Service

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

Parameters Defining QoS

- **Throughput/bandwidth -** the total amount of work completed during a specific time interval.
- **Delay** - the elapsed time from when a request is first submitted to and when the desired result is produced.
- **Jitter -** the delays that occur during the playback of a stream.
  - Due to lost frames.
  - Not acceptable for continuous media applications
  - Random variation of transmission time
- **Reliability/loss -** how errors are handled during transmission and processing of continuous media.

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

Stringency of applications' quality-of-service requirements.

Application Requirements (2)

- Bandwidth:
  - E-mail, audio, remote login not need much bandwidth. But file sharing and video in all forms need high bandwidth.
- Delay:
  - File transfer, email, audio, and video are not delay sensitive. That is if the packets are delayed uniformly across all by a few seconds, no harm is done. So, playing audio or video files from a server does not require low delay.
  - But, interactive applications, web access surfing and remote login are more delay sensitive. Telephony, and video conferencing are strictly delay sensitive
- Jitter
  - E-mail, file sharing and web access not sensitive to irregular arrival of packets.
  - Remote login is sensitive
  - Video and audio are extremely sensitive to Jitter
- Loss:
  - E-mail, file sharing, web access and remote login have more stringent requirements on loss and bits should be delivered correctly.
  - Audio and video applications tolerate lost packets.
    - People do not notice short pauses or skipped frames.
-

Application Requirements (3)

- Network support different categories of QoS
  - Constant bit rate (telephony)
    - Simulate a wire by providing uniform bandwidth and a uniform delay
  - Real-time variable bit rate (e.g., compressed video conferencing)
    - Sending a complex frame requires many bits, where a shot of white ball requires less bits.
  - Non-real-time variable bit rate (e.g., watching a movie on demand)
    - Buffering can be used
  - Available bit rate (e.g., file transfer)

    -Use the available bandwidth.

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)

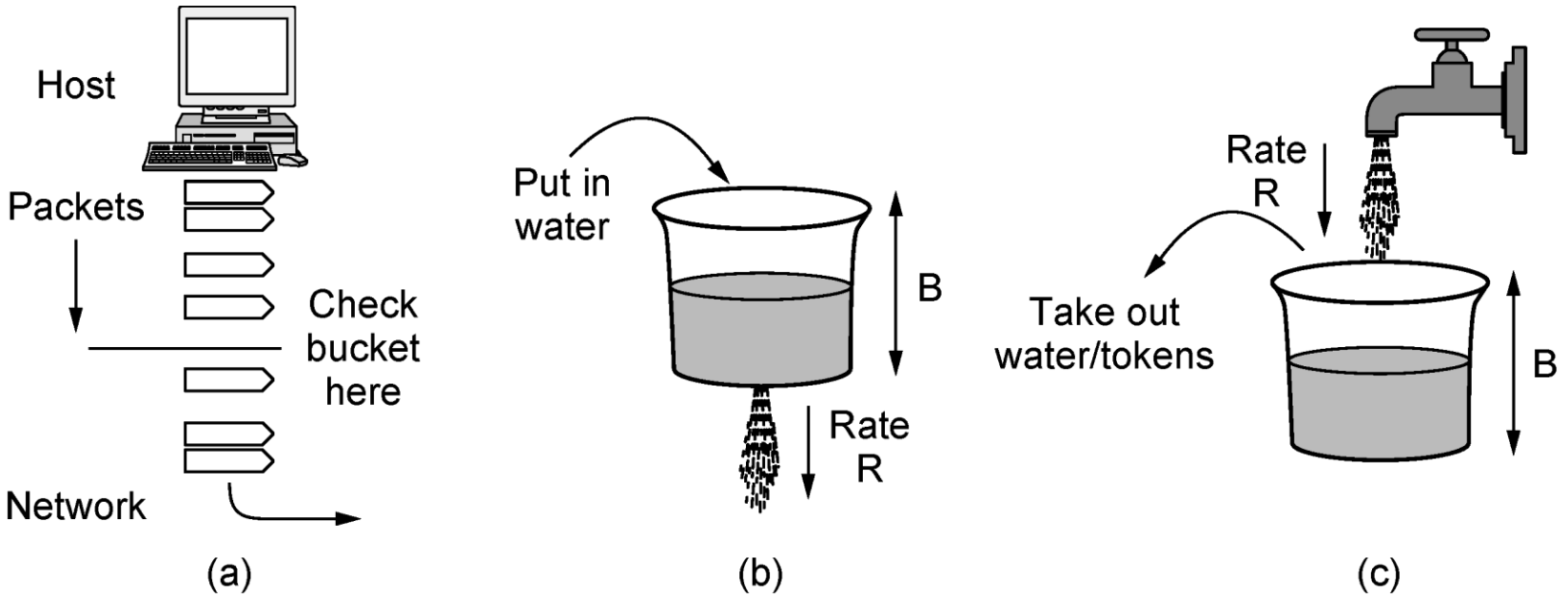
- We must know that traffic is being guaranteed
- However, traffic in data is BURSTY!
- Bursts in the traffic are difficult to handle than constant rate traffic as they can fill the buffers and cause packets to be lost.
- Traffic shaping
  - regulates the average rate and burstiness
- Monitoring traffic flow is called “Traffic policing”
- Traffic shaping and Traffic policing are important for real-time data (audio and video).

Leaky bucket algorithm

- Sliding window controls the data to be transmitted.

- Leaky bucket algorithm
  - Imagine a bucket with a small hole at the bottom.
  - No matter the rate the data enters, the outflow is constant.
  - Once the bucket is full with a capacity B, any additional water splits over the sides and lost.
  - Algorithm
    - Each host is connected to the network containing a leaky bucket.
    - If the packet arrives when the bucket is full, it is queued until enough water leaks out to hold it or be discarded.

Leaky and Token buckets



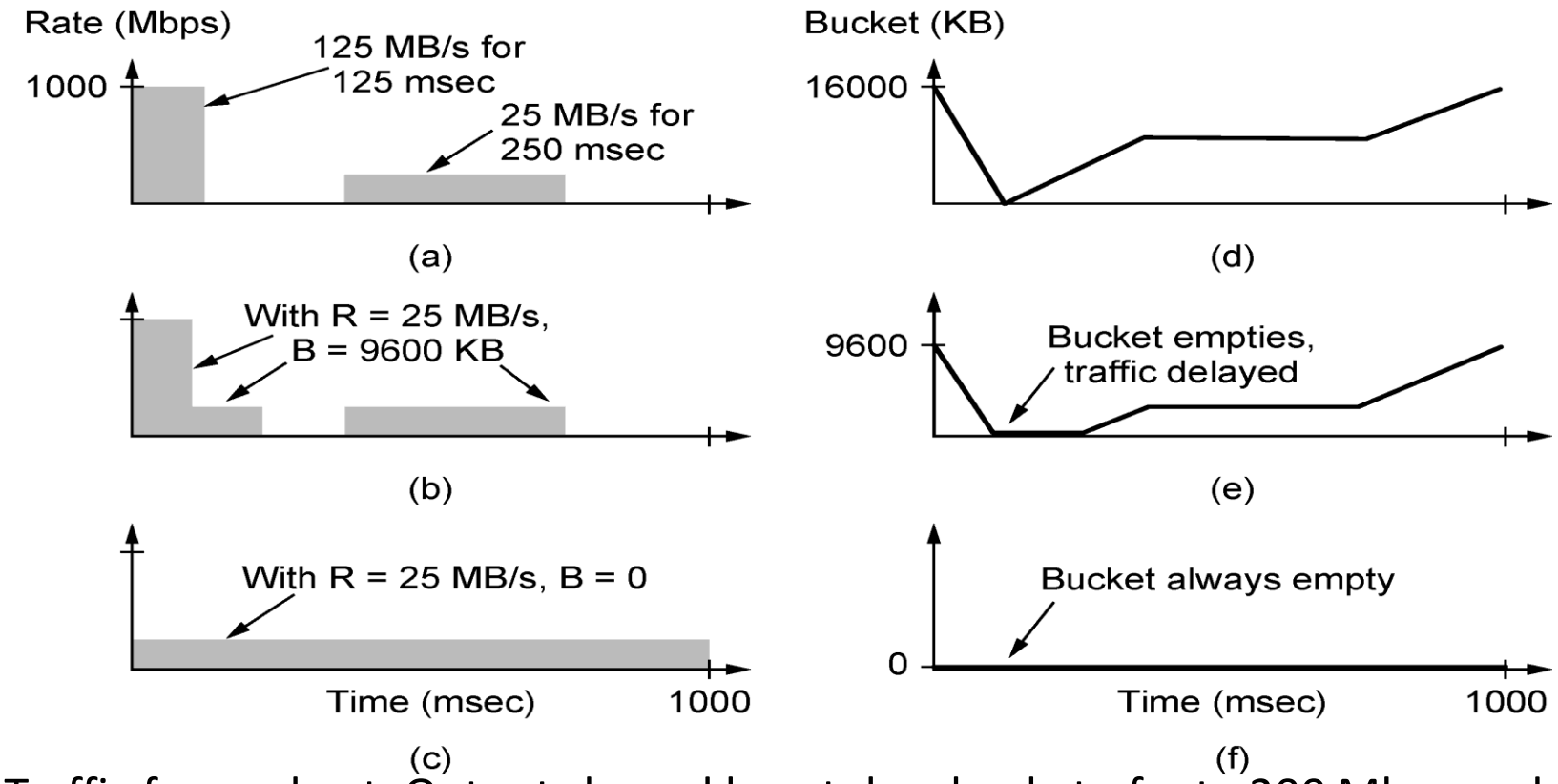
(a) Shaping packets. (b) A leaky bucket. (c) A token bucket.

Token bucket algorithm

- Imagine a bucket that is being filled. A tap is running at the rate R and the bucket has a capacity of B, as before.
- To send a packet, we must able to take water or tokens as out of bucket.
- If the bucket is empty, we must wait till tokens arrive.
- No more than a fixed number of tokens can accumulate in the bucket.
- Both leaky and token bucket algorithms smoothen the traffic by limiting the rate of flow.

Traffic Shaping (3)

- Both leaky and token bucket algorithms smoothen the traffic by limiting the rate of flow.



(a) Traffic from a host. Output shaped by a token bucket of rate 200 Mbps and capacity (b) 9600 KB and (c) 0 KB. Token bucket level for shaping with rate 200 Mbps and capacity (d) 16,000 KB, (e) 9600 KB, and (f) 0 KB.

- Host generates the traffic pattern as shown in Figure (a). The pattern in bursty.
  - Average rate over one second is 200Mbps, even though the host sends a burst of 16,000 KB at the top speed of 1000 Mbps (1/8 of second).
  - Routers can only accept data until only the buffers are filled. The buffer size is 9600 KB, smaller than the traffic burst. Some of the packets will be dropped.

- With token bucket, we can shape the traffic. The output of token bucket is shown in Figure (b). The host can send full throttle for a short while, until it has drained the bucket. It will cut back 200Mbps until the burst has been sent.
- The level of token bucket is given in Figure (e)
- We can also shape the traffic to be less busty. Figure (c) shows the output of a token bucket with R=200 Mbps and a capacity of zero. This is the extreme case. The corresponding bucket level Figure (f) is always empty.
- Figure (d) shows the bucket level for a token bucket with R=200 Mbps and a capacity of B=16,000 KB.
- This is the smallest token bucket through which the traffic passes unaltered.

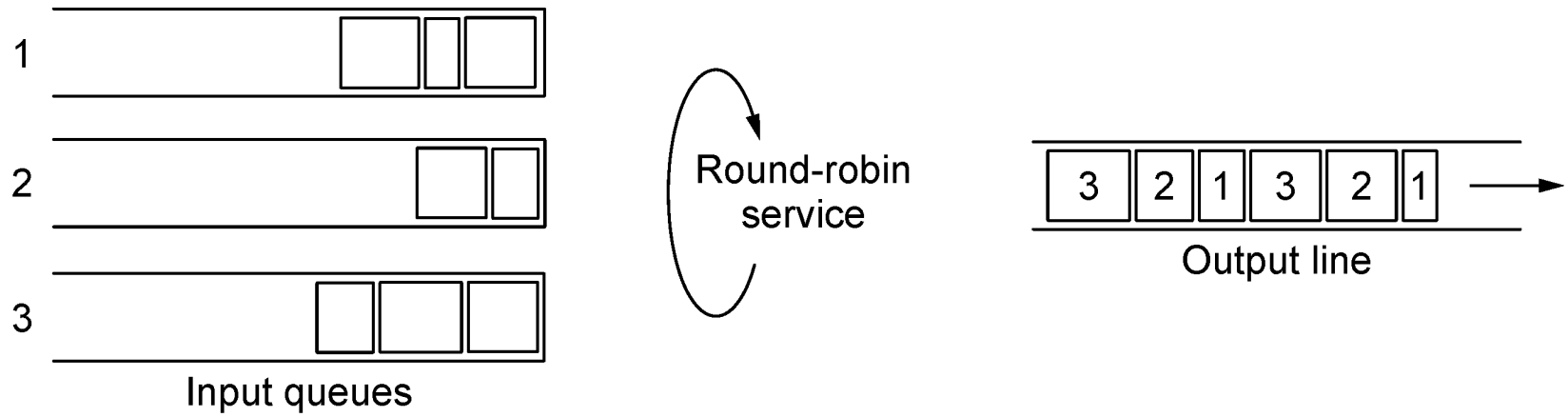
- Leaky and token buckets are easy to implement.

Packet Scheduling (1)

- Regulating the traffic is OK.
- However, to provide performance guarantee, we must reserve sufficient resources along the route for performance guarantee.
- Packet scheduling algorithms
  - Allocate router resources among the packets of a flow and between competing flows.
- Three kinds of resources are reserved
  - Bandwidth
    - If flow requires 1Mbps and bandwidth is 2Mbps, one should not try to get three flows through a line.
  - Buffer space
    - Purpose of buffer is to absorb the small bursts. To have quality of service, some buffers are reserved for a specific flow.
  - CPU cycles
    - Making sure that CPU is not loaded

Packet Scheduling (1)

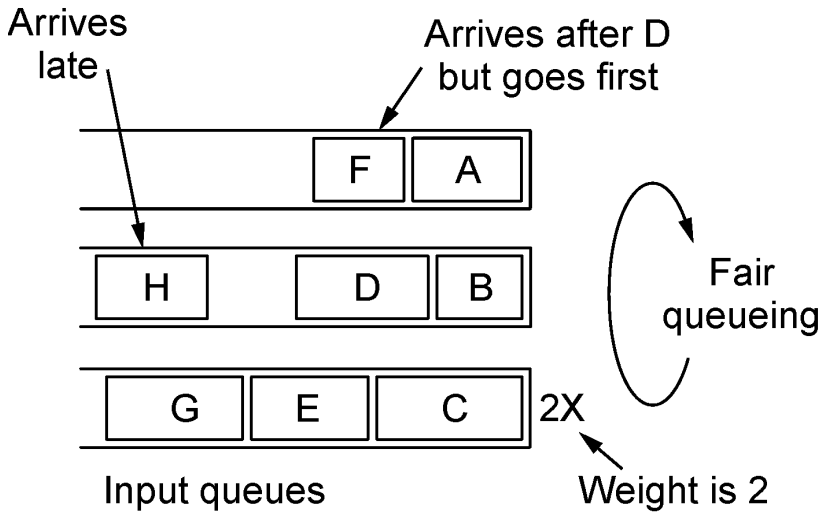
- FIFO: First-Come-First-Serve
  - Drop newly arrived packets
  - Tail drop behavior
  - Simple to implement
  - Not good for achieving quality of service in case of multiple flows
- Fair queuing round-robin
  - Routers have separate queues, one for each flow.
  - When line becomes idle, the router scans queues in a round robin fashion.



Round-robin fair queueing.

Packet Scheduling (2)

- Byte-byte round robin.
  - Compute the virtual time that is the number of round at which each packet would finish being sent.
  - It gives all hosts the same priority
- Weighted Fair Scheduling (WFG)
  - Number of bytes per round will be weight (W) of the flow.
  - Fi=max(Ai,Fi-1)+Li/W
    - Ai is the arrival time, Fi is the finish time, Li is the length of the packet



(a)

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

(b)

(a) Weighted Fair Queueing. (b) Finishing times for the packets.

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

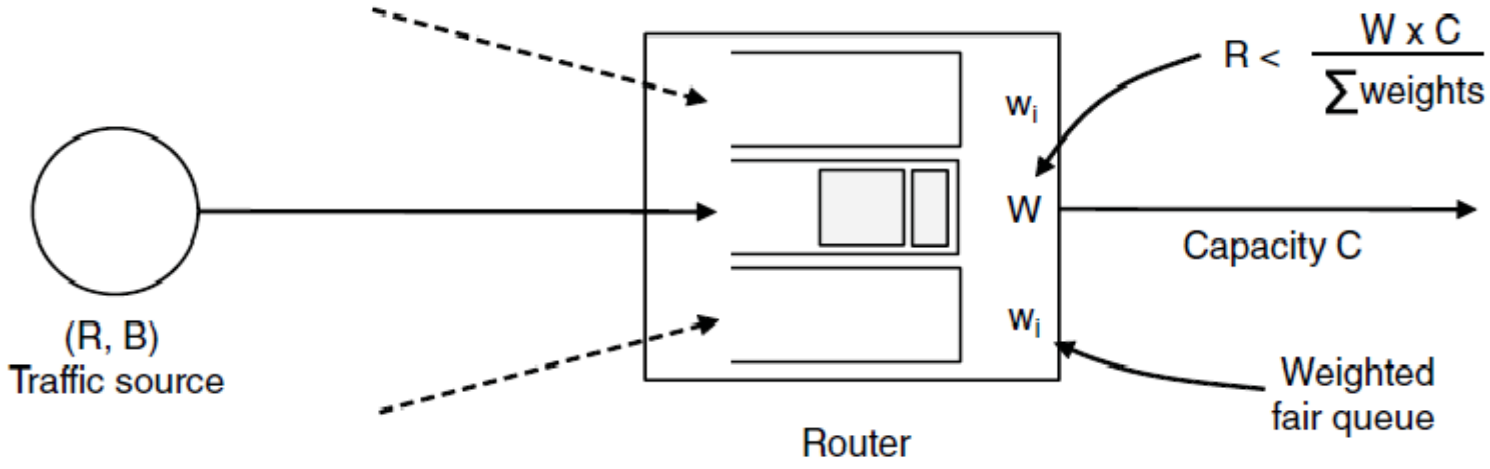
Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

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 (Waited Fair Queue) with weight W / all weights > R/capacity
- Holds for all traffic patterns, all topologies



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 »

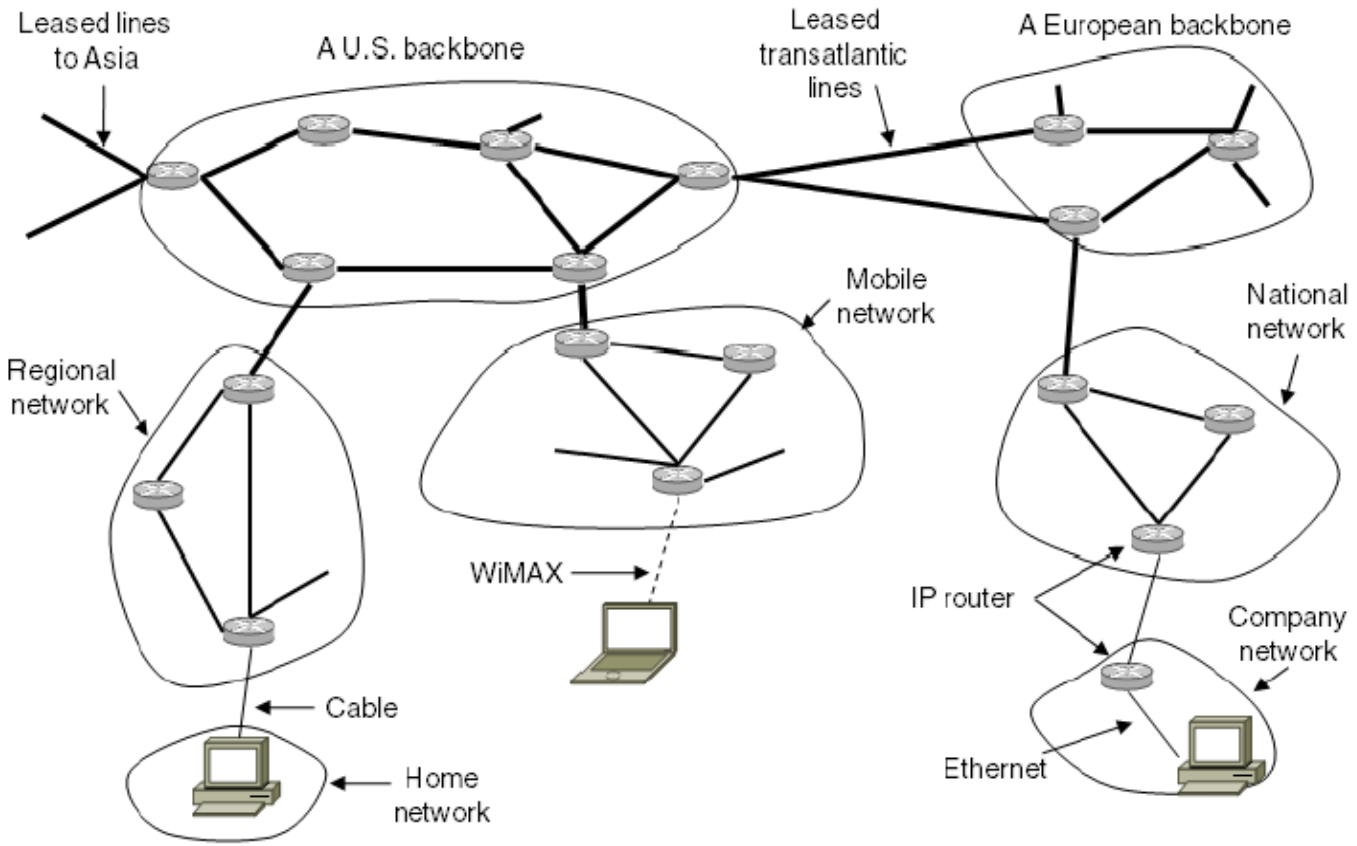
Network Layer in the Internet (2)

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

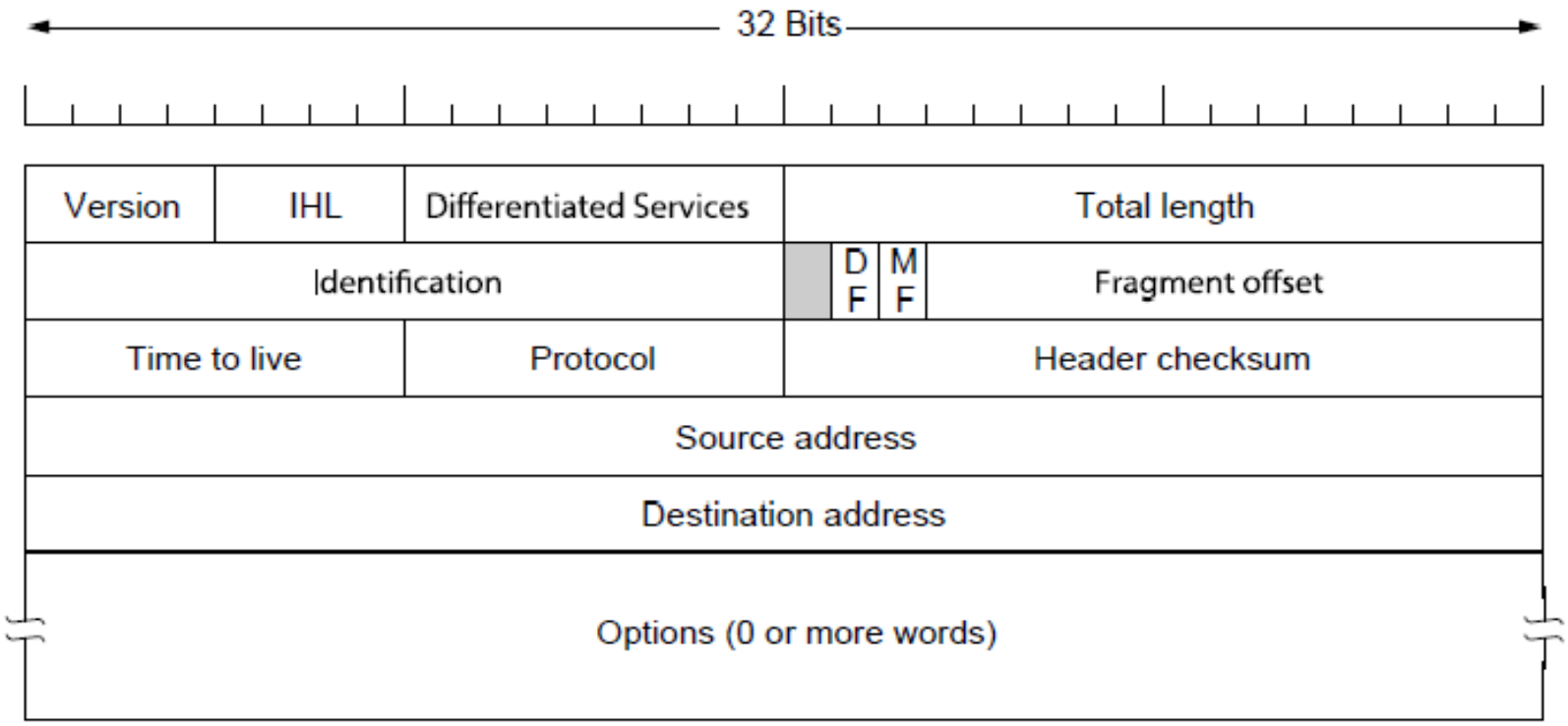
Network Layer in the Internet (3)

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:



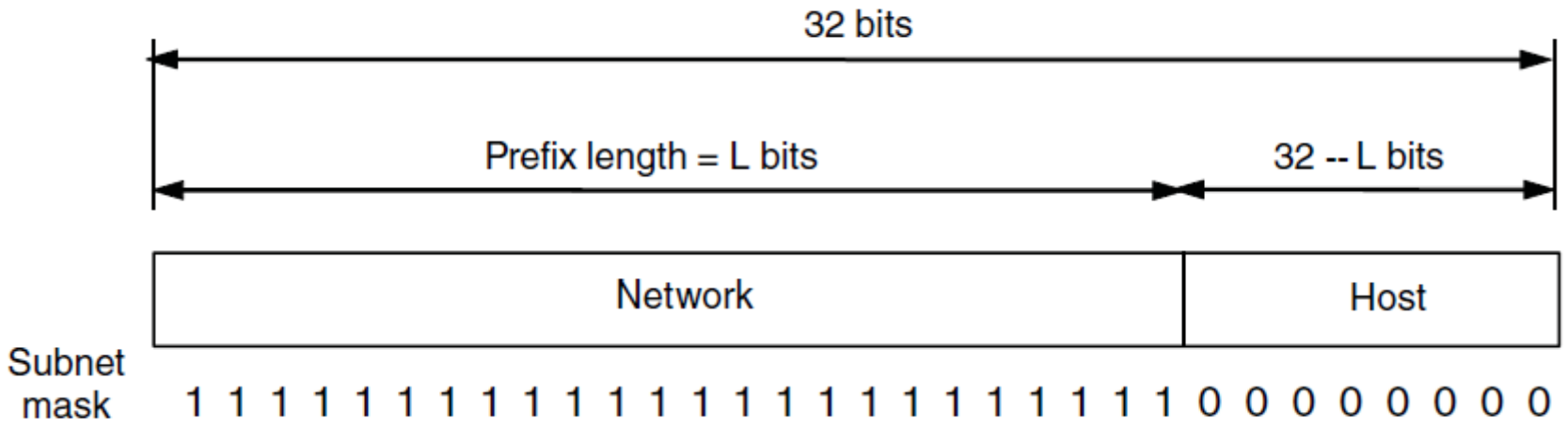
IP Version 4 Protocol (1)

- Version filed
- IHL: header length
- Differentiated services: type of service
- Total length: every thing in the datagram
- Identification field: which packet the newly arrived fragment belong to
  - All the fragments of the same packet have the same identification code.
- Next field: not defined
- DF: do not fragment
- MF more fragments
  - It is set to know whether all the fragments of datagram have arrived.
- Fragment offset
  - Where in the current packet this packet belongs
- TtL: Time to live
  - To limit packet life time
  - Decremented after each hop
- Protocol: UDP or TCP
- Header checksum
- Source and destination address indicate the IP address of source and destination
- Options field
  - For other ideas
  - Security
  - Strict route
    - Set of routes
  - Lose routes
    - Set of routes

IP Addresses (1) – Prefixes

Addresses are allocated in blocks called prefixes

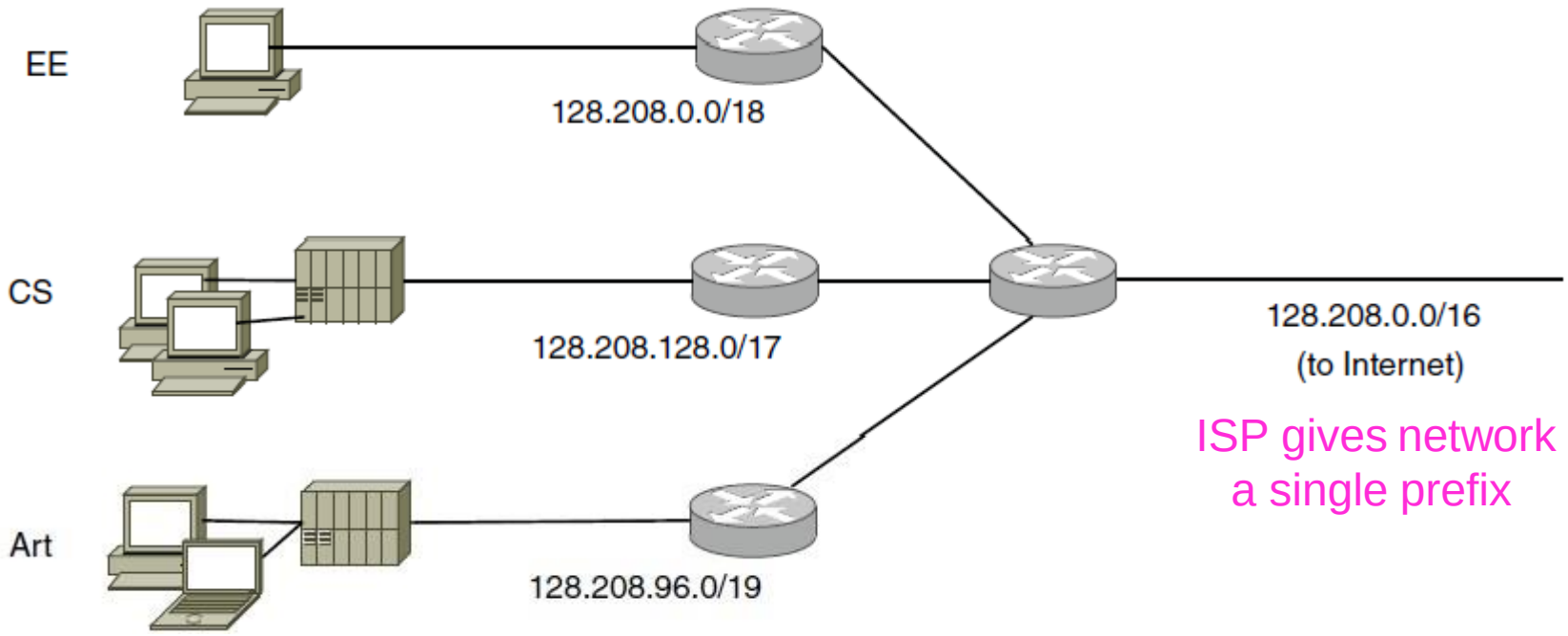
- Prefix is determined by the network portion
- Has 2^L addresses aligned on 2^L boundary
- Written address/length, e.g., 18.0.31.0/24



IP Addresses (2) – Subnets

Subnetting splits up IP prefix to help with management

- Looks like a single prefix outside the network

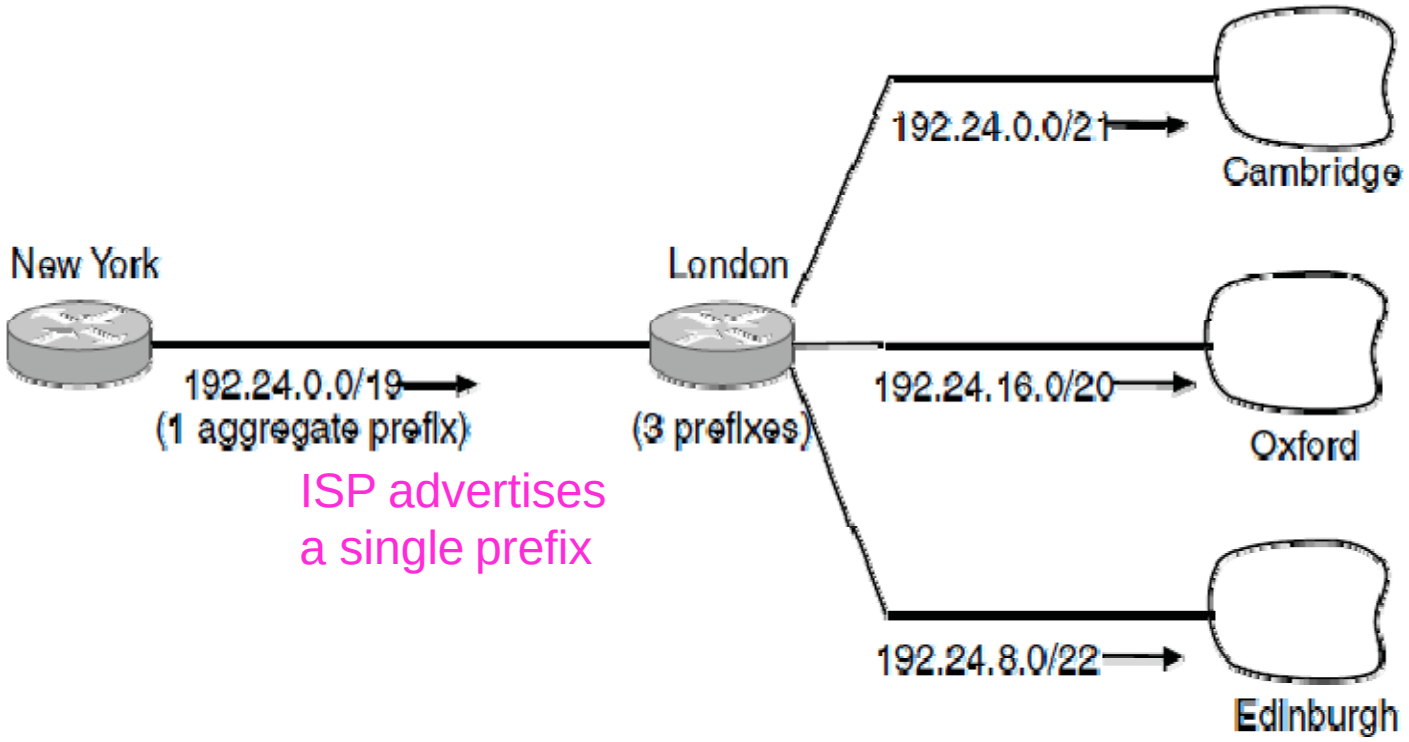


ISP gives network a single prefix

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



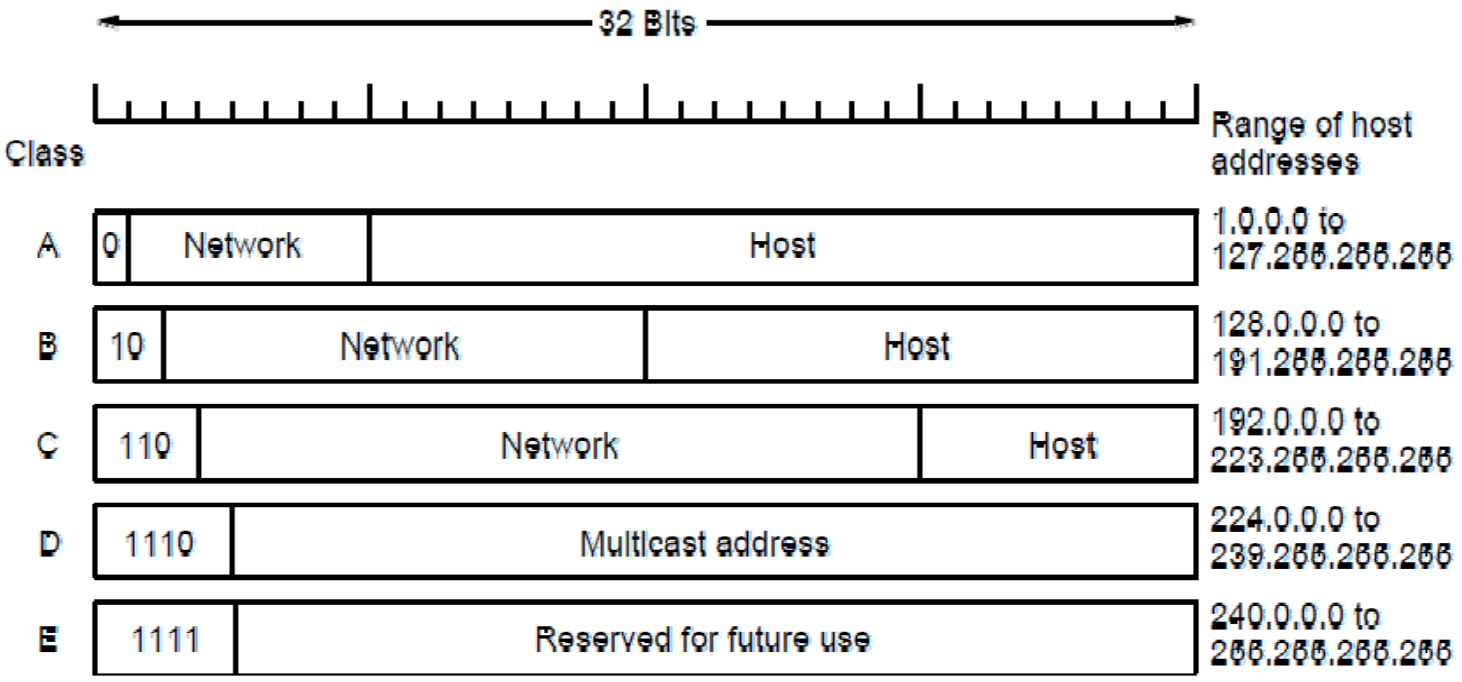
ISP advertises a single prefix

ISP customers have different prefixes

IP Addresses (5) – Classful Addresing

Old addresses came in blocks of fixed size (A, B, C)

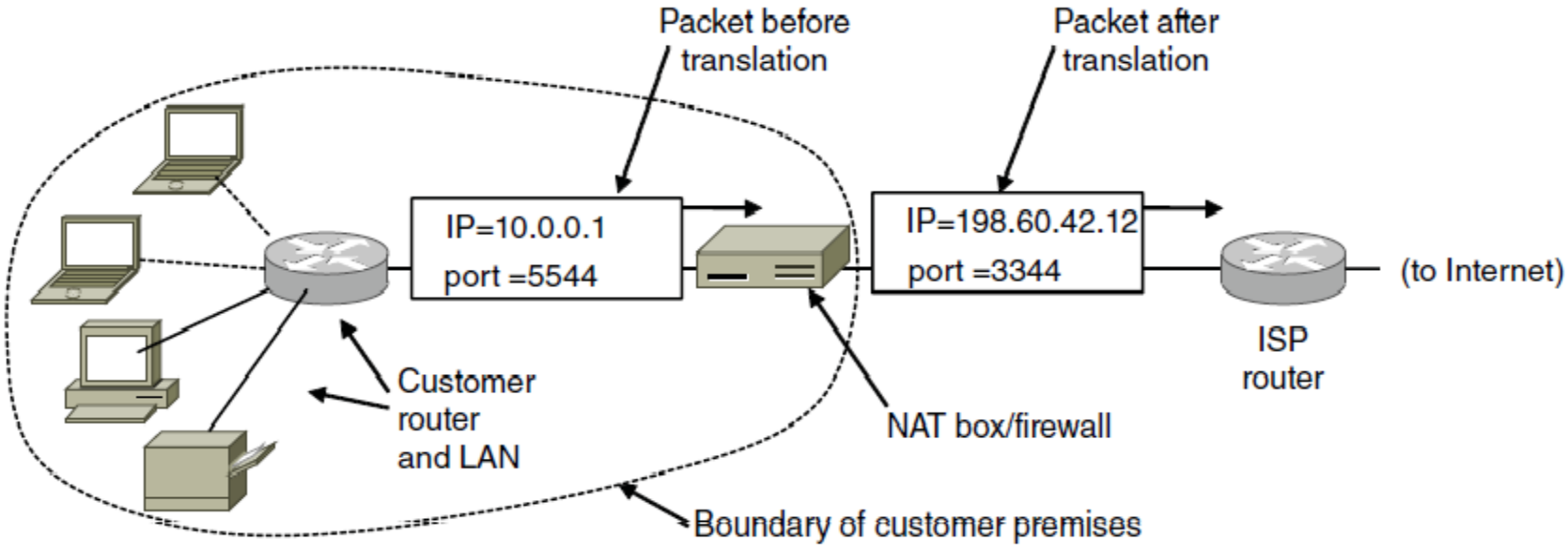
- Carries size as part of address, but lacks flexibility
- Called classful (vs. classless) addressing



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.



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

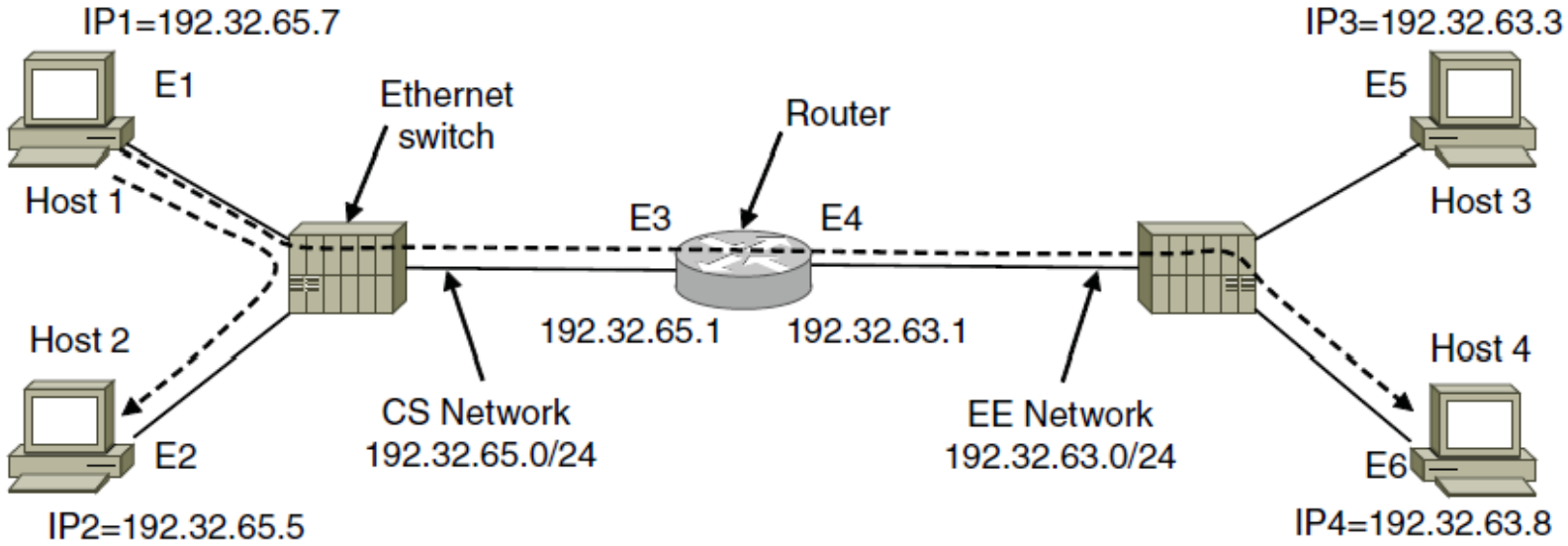
Internet Control Protocols (2)

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

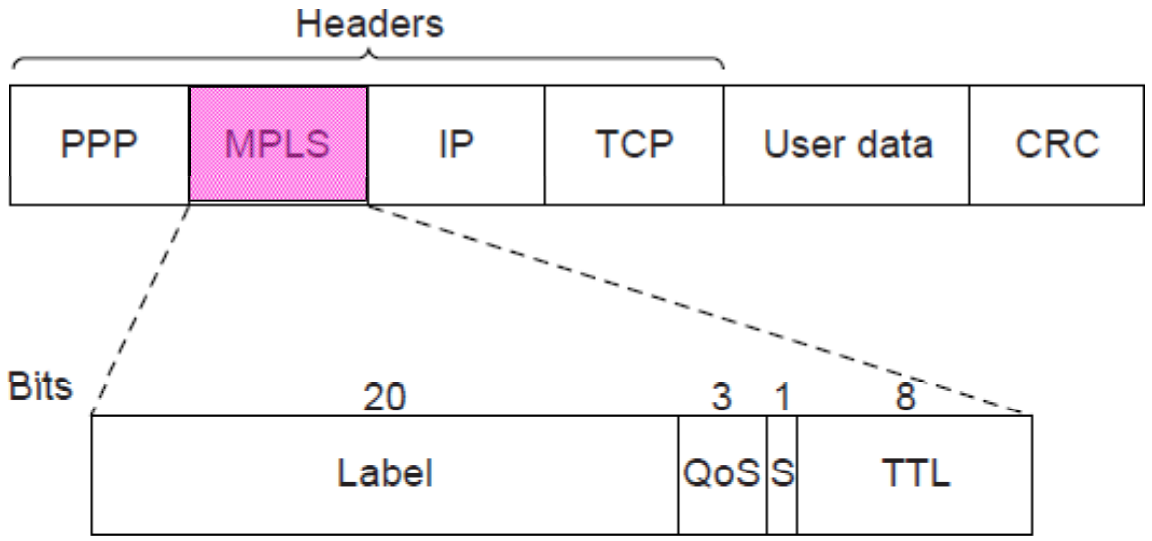


Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

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



End

Chapter 5