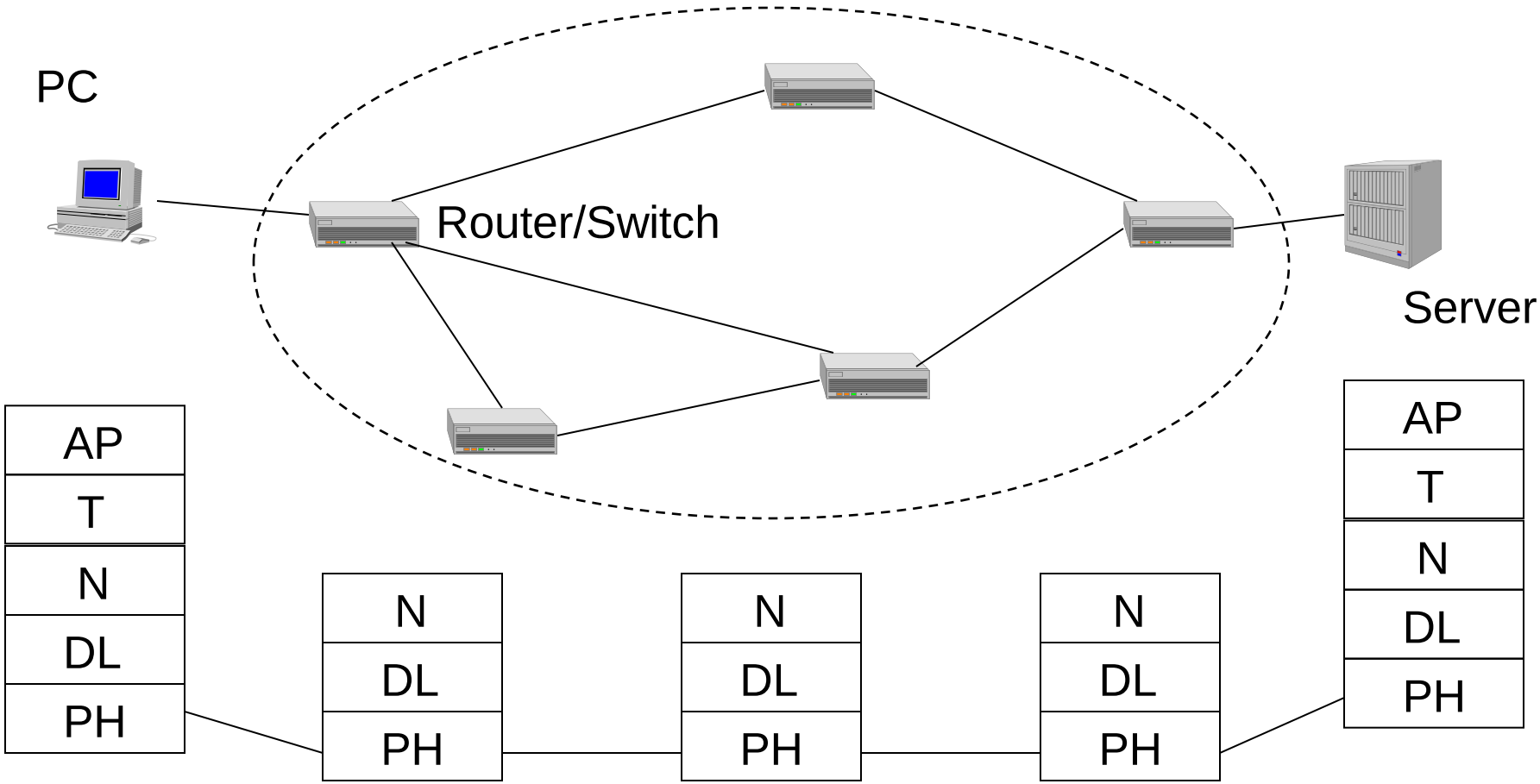
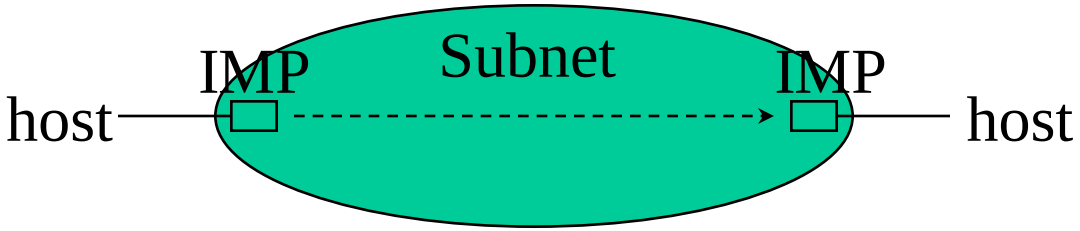
The Network Layer

The Network Layer



The Network Layer (cont'd)

- Objective: getting packets from the source all the way to the destination of the subnet



Main Tasks of the Network Layer

- Providing services to the higher layer protocol
- Addressing
- Routing
- Congestion Control
- Internetworking
- Accounting

Services Provided to the User

Services perceived by the user applications can be categorized as:

- Connectionless service
  - network is assumed to be unreliable
  - no connection setup prior to data exchange
  - applications need to handle packet ordering, error control, flow control, etc
  - for example, UDP

⇨ Complexity is placed on the host.

Services Provided to the User (cont'd)

- Connection-oriented Service
  - network should provide a reliable service
  - a connection is set up first and the two end points can negotiate about the parameters
  - packets are delivered in order and error-free. Flow control is automatic
  - for example, TCP

⇨ Complexity is placed on the network.

Routing

- Combinations of service and subnet structure

Upper layer	Type of subnet: Datagram	Type of subnet: Virtual circuit
Connectionless	UDP over IP	UDP over IP over ATM
Connection-oriented	TCP over IP	ATM AAL1 over ATM

Fig. 5-3. Examples of different combinations of service and subnet structure.

- A major function of the network layer
- Invoked at call set up time for the VC service
- Invoked for every packet for the datagram service

Routing (cont'd)

- Desired properties for routing
  - correctness
  - simplicity
  - robustness (to cope with topology and traffic changes)
  - stability (to converge to equilibrium)
  - optimality
  - fairness

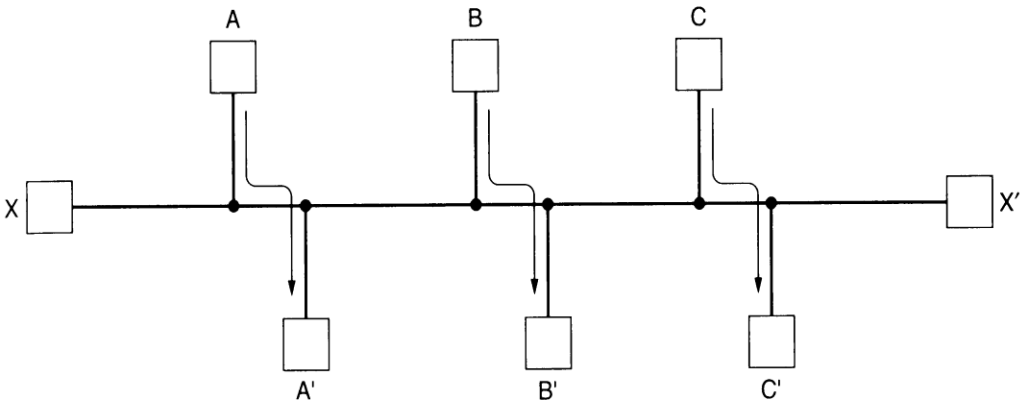


Fig. 5-4. Conflict between fairness and optimality.

Routing (cont'd)

- Nonadaptive (static) vs. adaptive routing
- Optimality principle: If router *J* is on the optimal path from router *I* to router *K*, then the optimal path from *J* to *K* also falls along the same path.

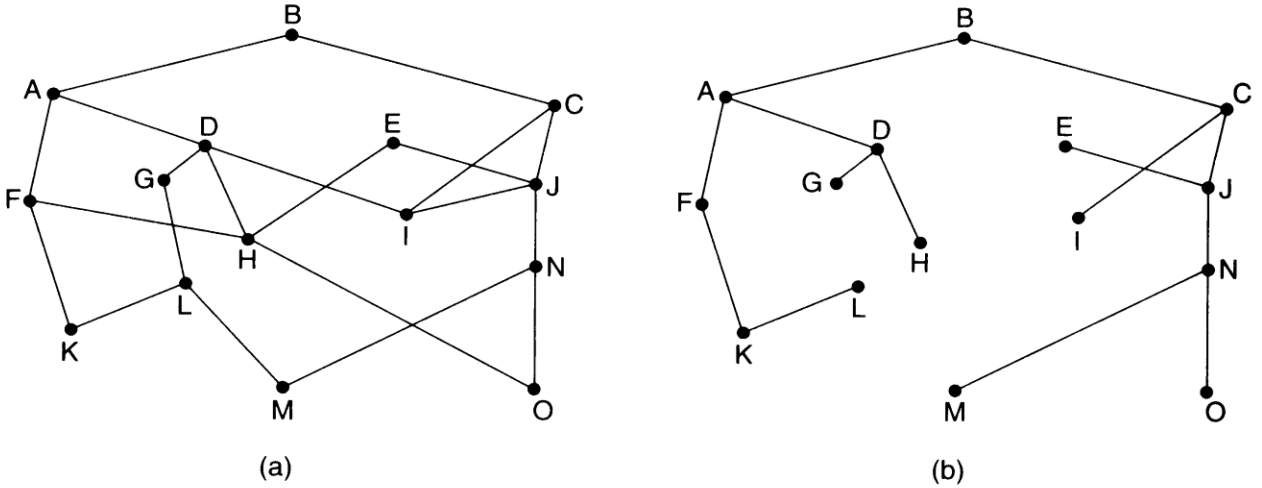
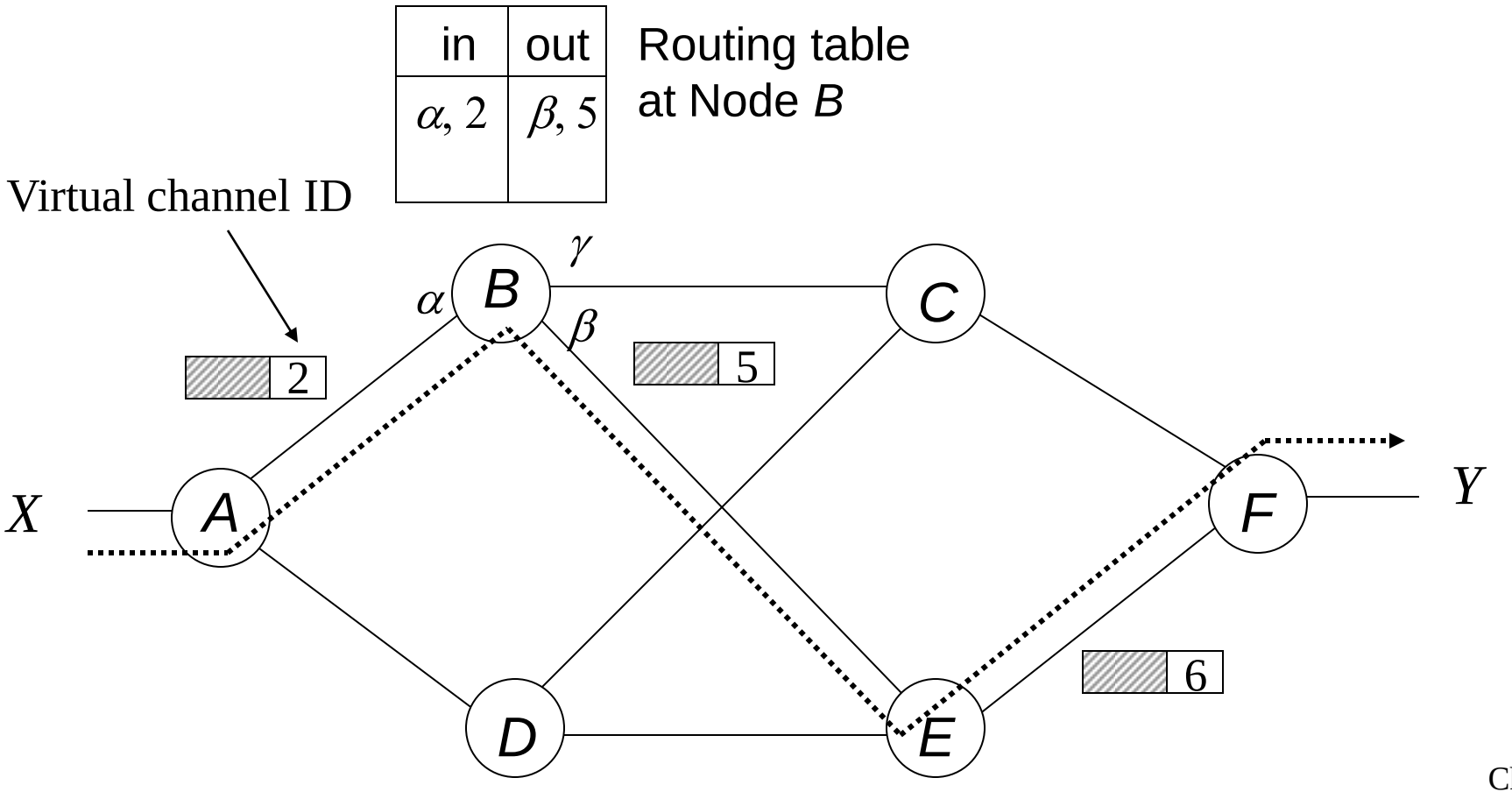


Fig. 5-5. (a) A subnet. (b) A sink tree for router *B*.

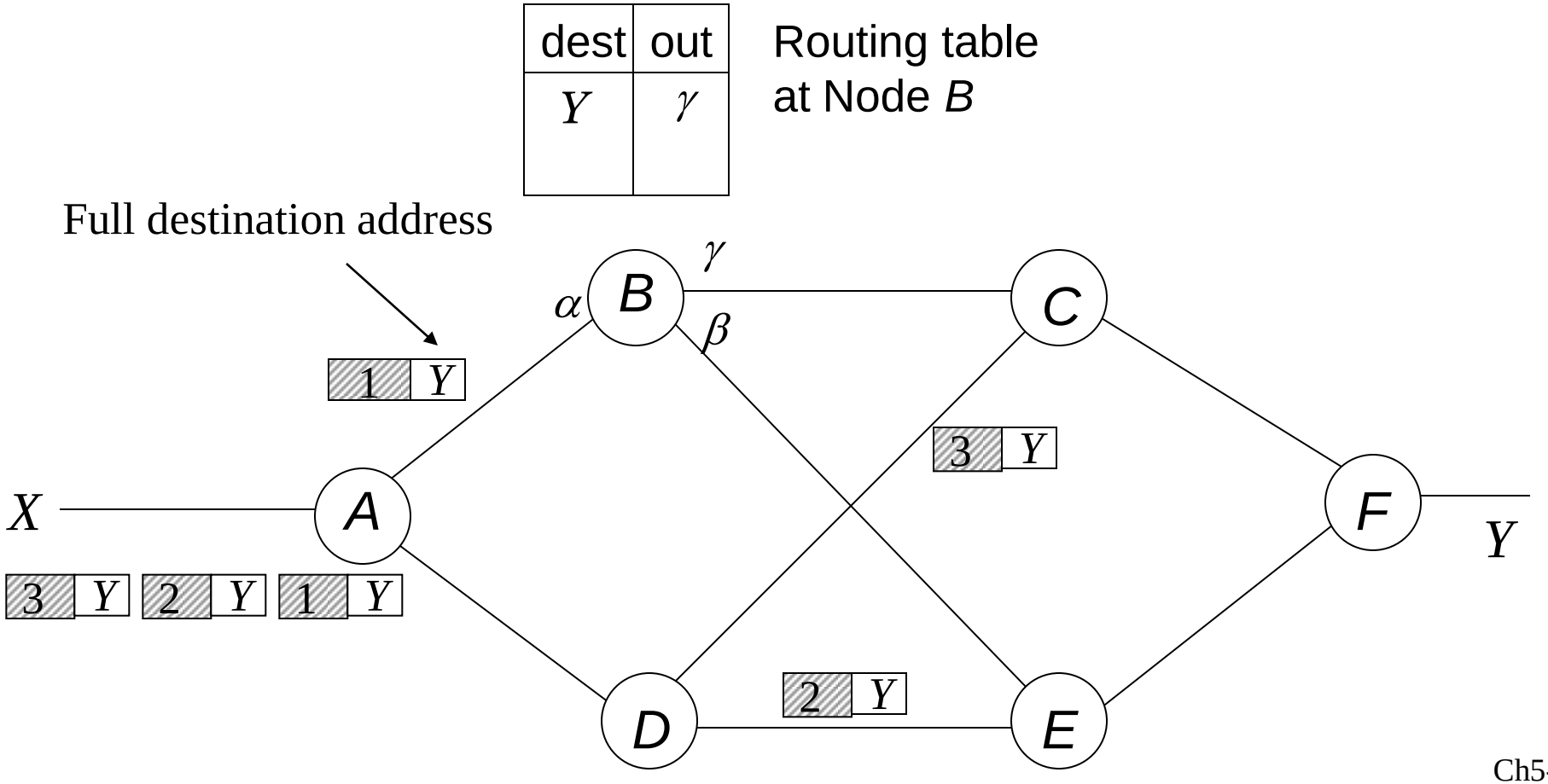
Virtual Circuit (VC) Routing

- Connection-oriented routing



Datagram (DG) Routing

- Connectionless routing



Comparisons of VC and DG

	VIRTUAL CIRCUIT	DATAGRAM
Table space	Required to store VC information	Does not need to store connection information
Packet header	Short	Long
Routing decision	Only during circuit setup	Calculated for every packet
Circuit setup delay	Yes	No
Congestion control	Easy	Difficult
Router failure effect	More serious	Less serious

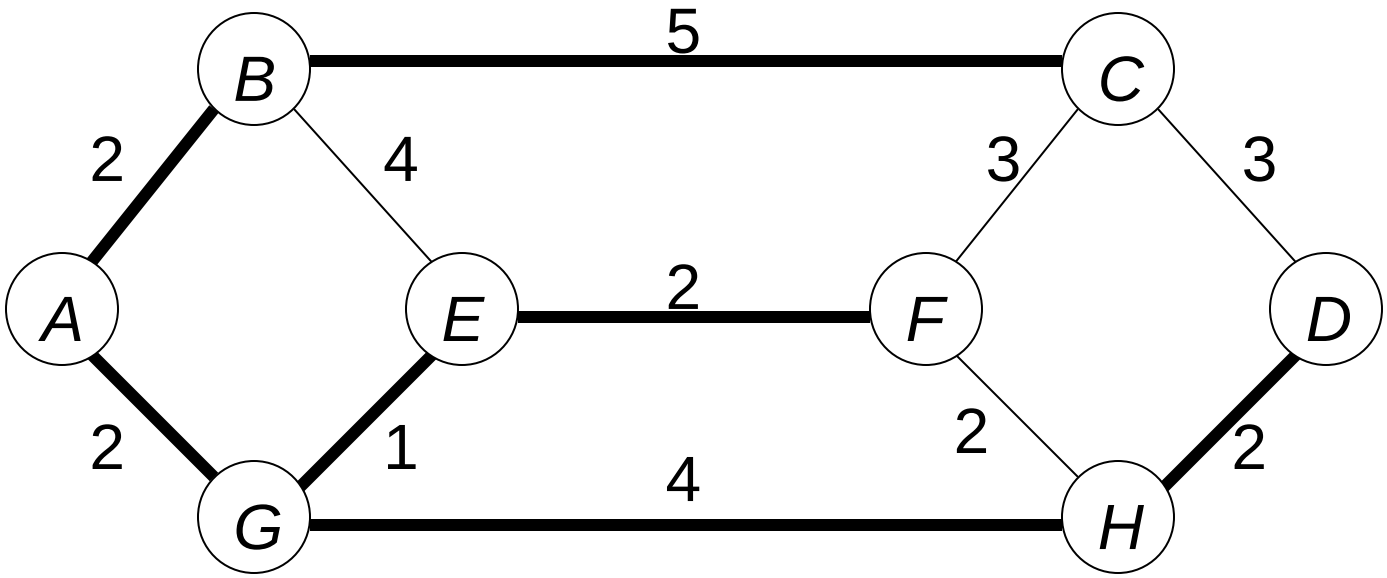
Shortest Path Routing

- Given: a graph of nodes (set *N*) and links (arcs) with associated arc weights (metrics), e.g. queue length, distance, delay and loss
- For each origin-destination (O-D) pair find a path with the minimum total arc weights along the path
- Centralized vs. distributed routing

Shortest Path Routing (cont'd)

Routing table at Node *A*

dest	next
B	*B*
C	*B*
D	*G*
E	*G*
F	*G*
G	*G*
H	*G*



- Each node computes the shortest paths to every other node in the network.
- The metric of a link can be distance, delay, hop, bandwidth, or combinations of them.

Shortest Path Routing (cont'd)

- Dijkstra's algorithm (to calculate a shortest path spanning tree rooted at node *r*)
  0. $S=\{r\}$, dist(i)=infinity for all i in N, dist(r)=0, $l=r$.
  1. dist(i)=min{dist(i), dist(l)+cost(l,i)} for every neighbor of l where i is not in S.
  2. Find among the nodes not in S a node with the minimum distance from r. Denote this node by l.
  3. $S=S$ unions $\{l\}$.
  4. If $S=N$, stop; otherwise, go to Step 1.

Shortest Path Routing (cont'd)

- Dijkstra's algorithm (cont'd)

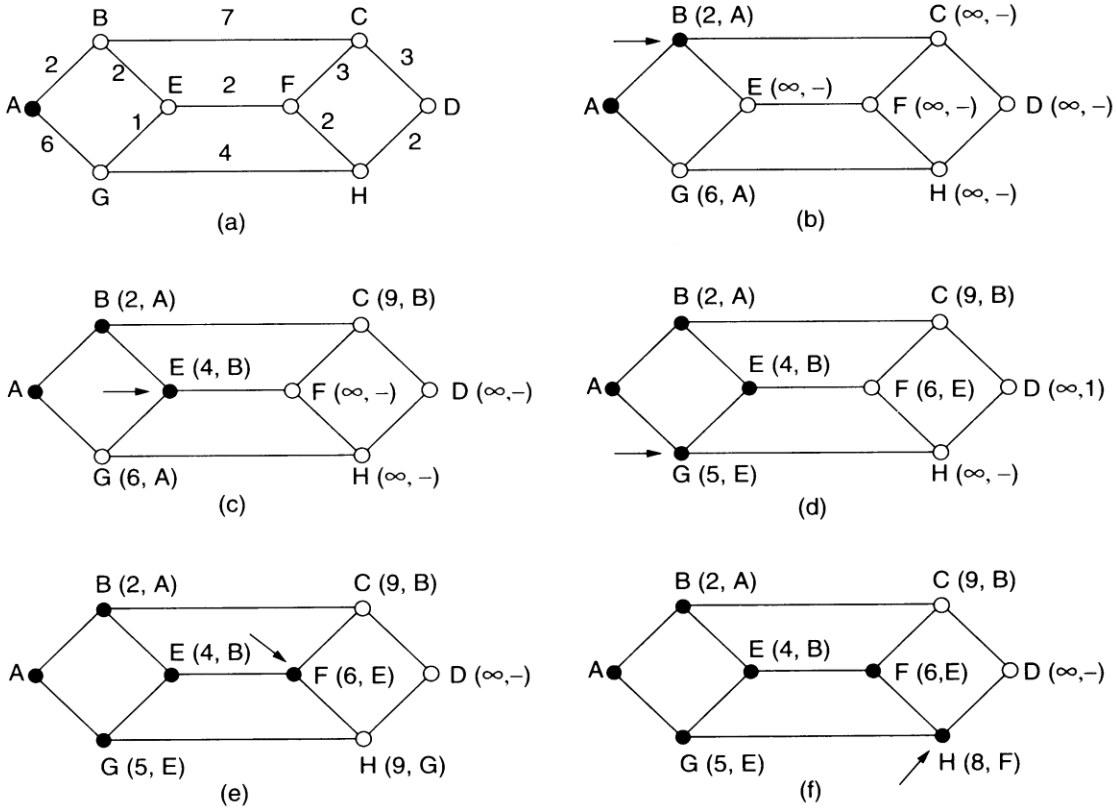
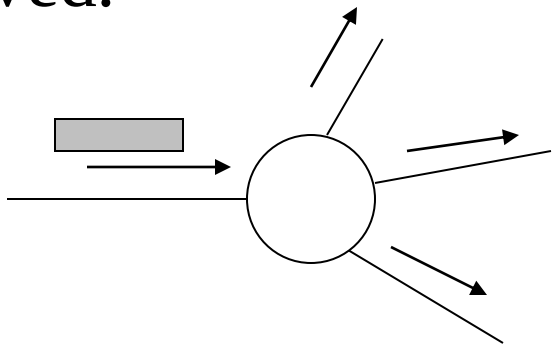


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Flooding

- When a router receives a packet, the router duplicates the packet and broadcast it to all the links except the one from which the packet was received.



- Flooding can be used to
  - discover all the routes between two points
  - exchange information network-wide

Flooding (cont'd)

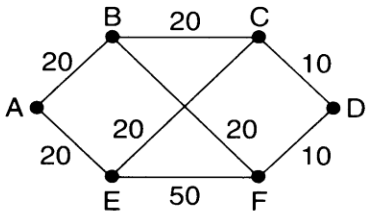
- Flooding will generate a vast number of duplicate packets.
- Several ways to control flooding:
  - hop counter in each packet: packet is discarded when the counter is decremented to zero
  - maintain list of packets that have already been seen
  - selective flooding: only duplicate and send to those lines that could be right

Flow-based Routing

- (Quasi-) static, capacitated and load sensitive
- Given
  - topology
  - link capacities
  - traffic requirement (data rate for each O-D pair)
- To determine: an optimal routing assignment
- Objective: to optimize a certain performance measure, e.g. to minimize the average end-to-end packet delay
- Subject to: multicommodity flow, nonnegativity and capacity constraints

Flow-based Routing (cont'd)

- An example of evaluating the average end-to-end packet delay using *M*/*M*/1 queueing models



(a)

Destination

Source	A	B	C	D	E	F
A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
B	9 BA		8 BC	3 BFD	2 BFE	4 BF
C	4 CBA	8 CB		3 CD	3 CE	2 CEF
D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
E	7 EA	2 EFB	3 EC	3 ECD		5 EF
F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

(b)

Fig. 5-8. (a) A subnet with line capacities shown in kbps. (b) The traffic in packets/sec and the routing matrix.

Flow-based Routing (cont'd)

- An example of evaluating the average end-to-end packet delay using *M*/*M*/1 queueing models (cont'd)

i	Line	λ_i (pkts/sec)	C_i (kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

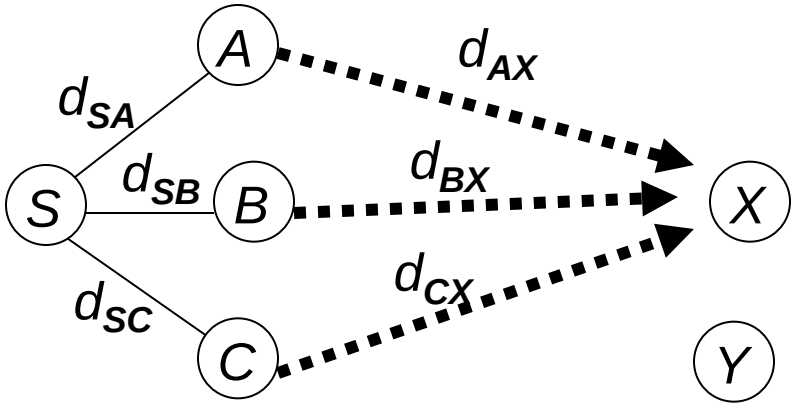
Fig. 5-9. Analysis of the subnet of Fig. 5-8 using a mean packet size of 800 bits. The reverse traffic (*BA*, *CB*, etc.) is the same as the forward traffic.

Distance Vector Routing

- Also called Bellman-Ford or RIP

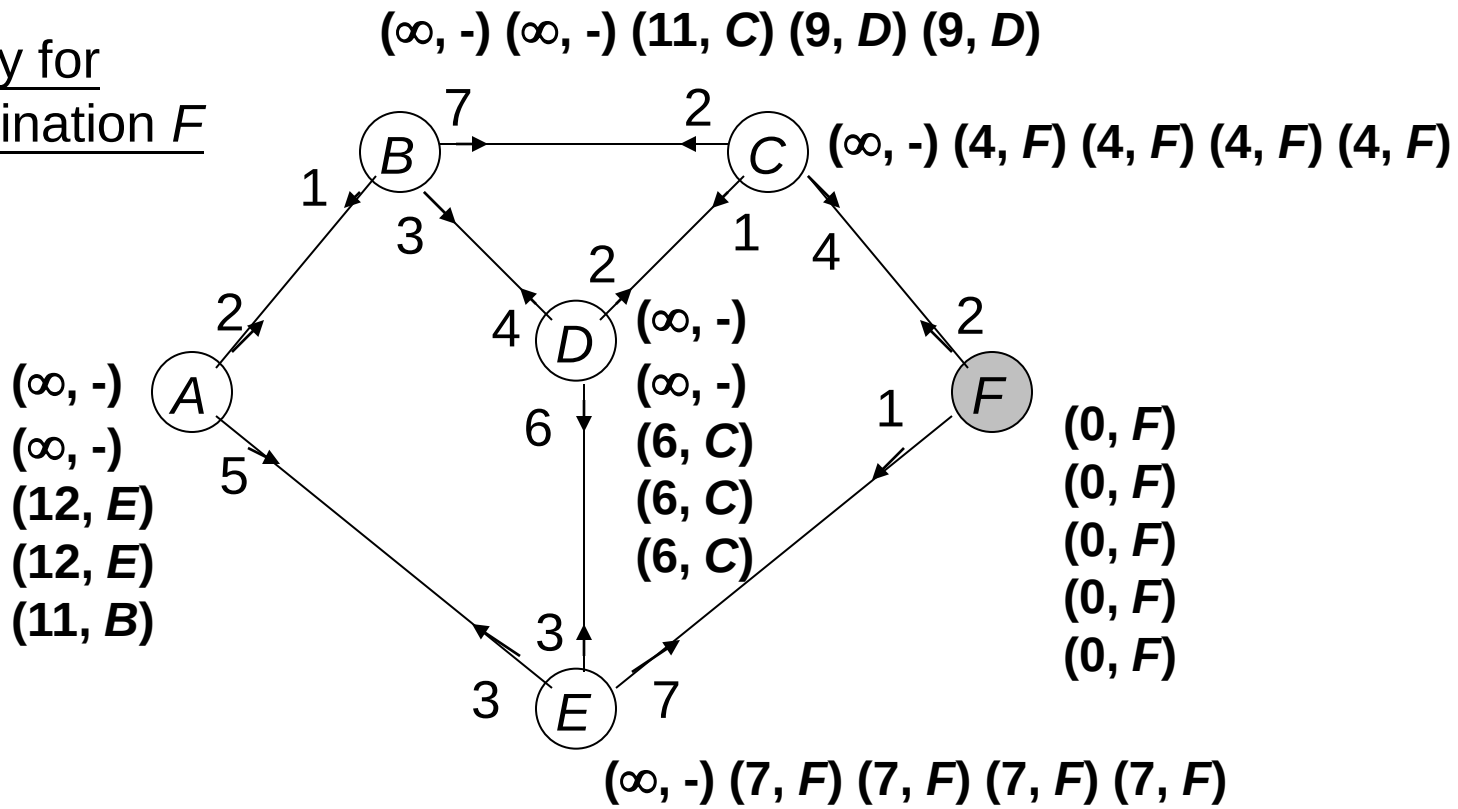
Routing table at Node *S*

Destination	Distance	Next
X	120	*A*
Y	100	*C*



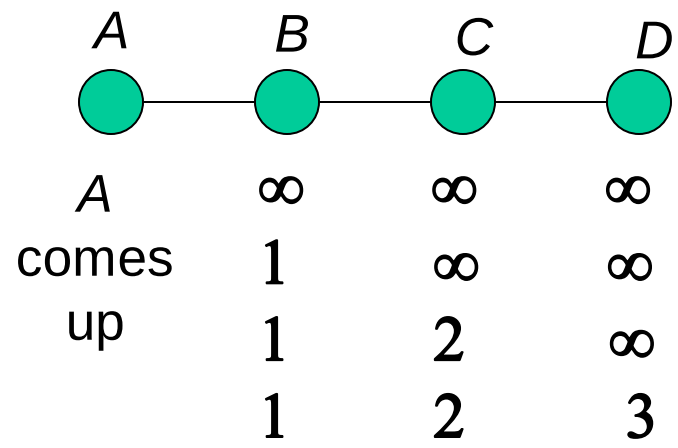
- Each router keeps monitoring distances (current queue length) to its direct neighbors
- Once every *T* sec it exchanges the (Destination, Distance) vector with all its neighbors
- New distance from *S* to *X* via Node *i* = $d_{si} + d_{ix}$
- Store the *i* that gives the minimum distance

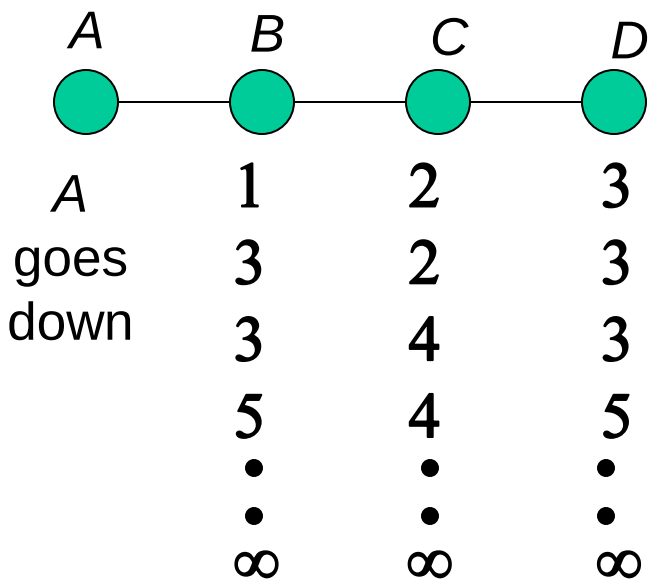
Distance Vector Routing Example



Note: At lease *N* updates are required to reach steady state, where *N* = network diameter

Count-to-Infinity Problem

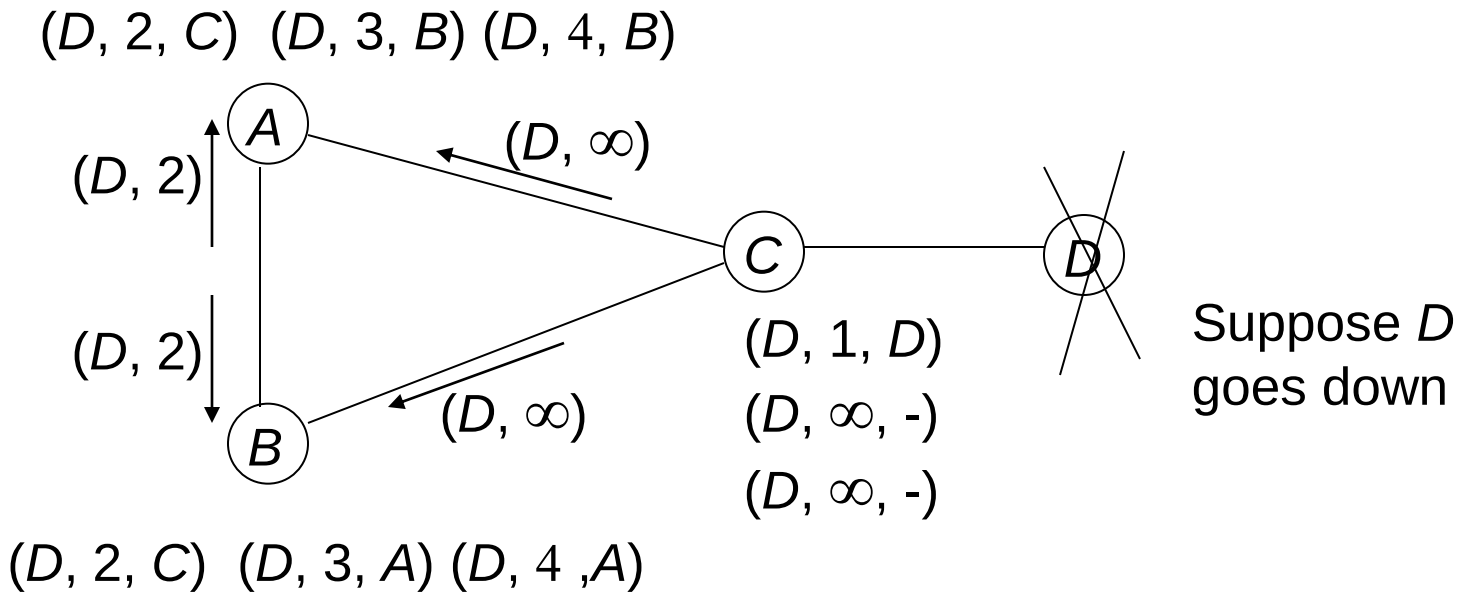




<u>One Solution -- Split Horizon algorithm:</u>

The distance to destination *X* is not reported to the neighbor which is the next hop for the packets destined to *X*

Ping-Pong Effect



Packets for *D* will be bounced back-and-forth between *A* and *B*.

Link State Routing

- OSPF, IS-IS are based on link state routing.

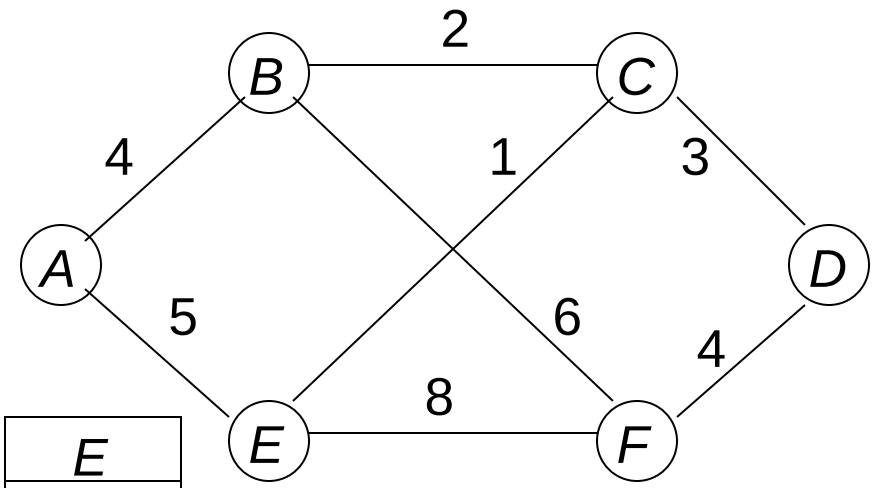
Link state routing has five steps:

- Discovering the neighbors
  - a just booted router sends HELLO packet on each link it connects
  - its neighbors reply with their names
- Measuring link delays
  - send ECHO packet to each neighbor and record how soon the reply comes back

Link State Routing (cont'd)

- Building link state packets every *T* seconds

A	
Seq #	
Age	
B	4
E	5



E	
Seq #	
Age	
A	5
C	1
F	8

Link State Routing (cont'd)

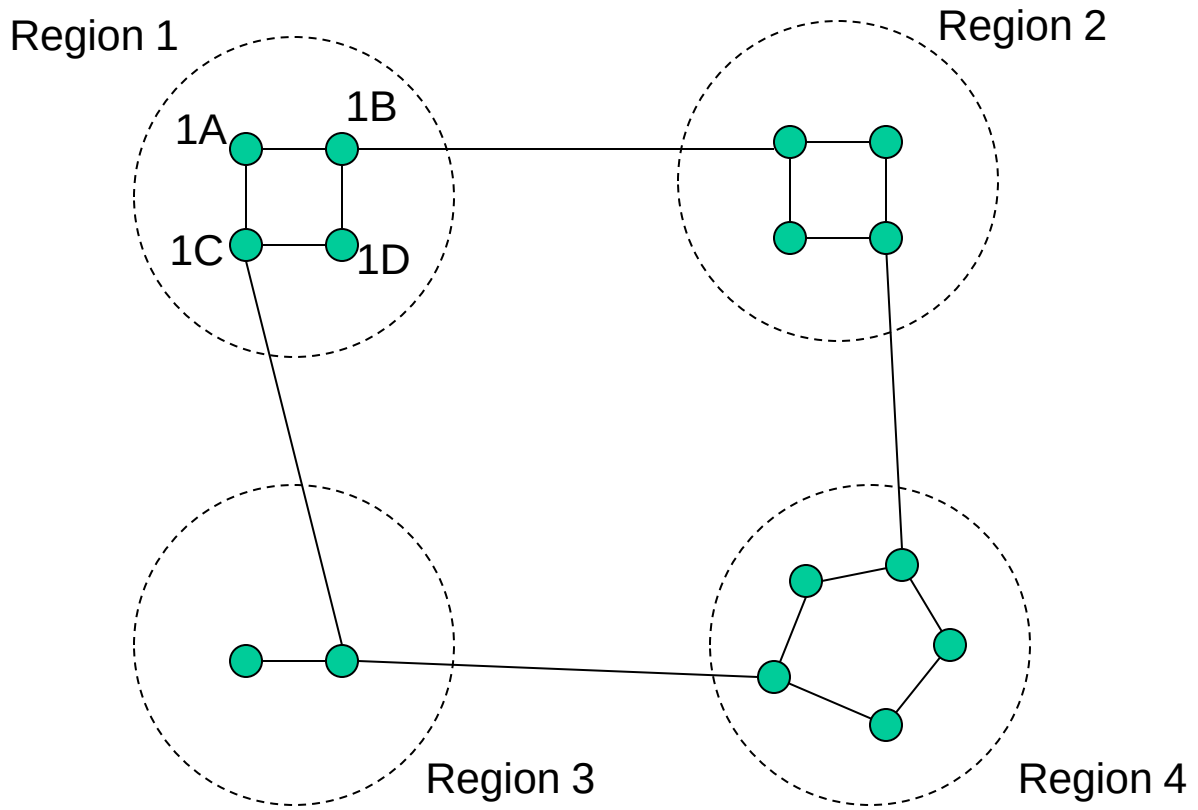
- Distributing the link state packets by flooding
  - source increments the seq# for each new packet
  - when a router receives a packet, check its (source, seq#)
    - duplicate packet is discarded
    - new packet is broadcast to all the lines except the incoming one
  - age: decremented by each router. The packet is discarded when age goes to 0
- Computing the new routes
  - each node constructs the entire network topology, and then
  - computes the shortest paths to all possible destinations

Hierarchical Routing

- The network is divided into hierarchies to reduce the size of the routing table

Routing table at Node 1A

Dest	Distance	Next
1B	60	1B
1C	80	1C
1D	110	1B
2	150	1B
3	180	1C
4	220	1C

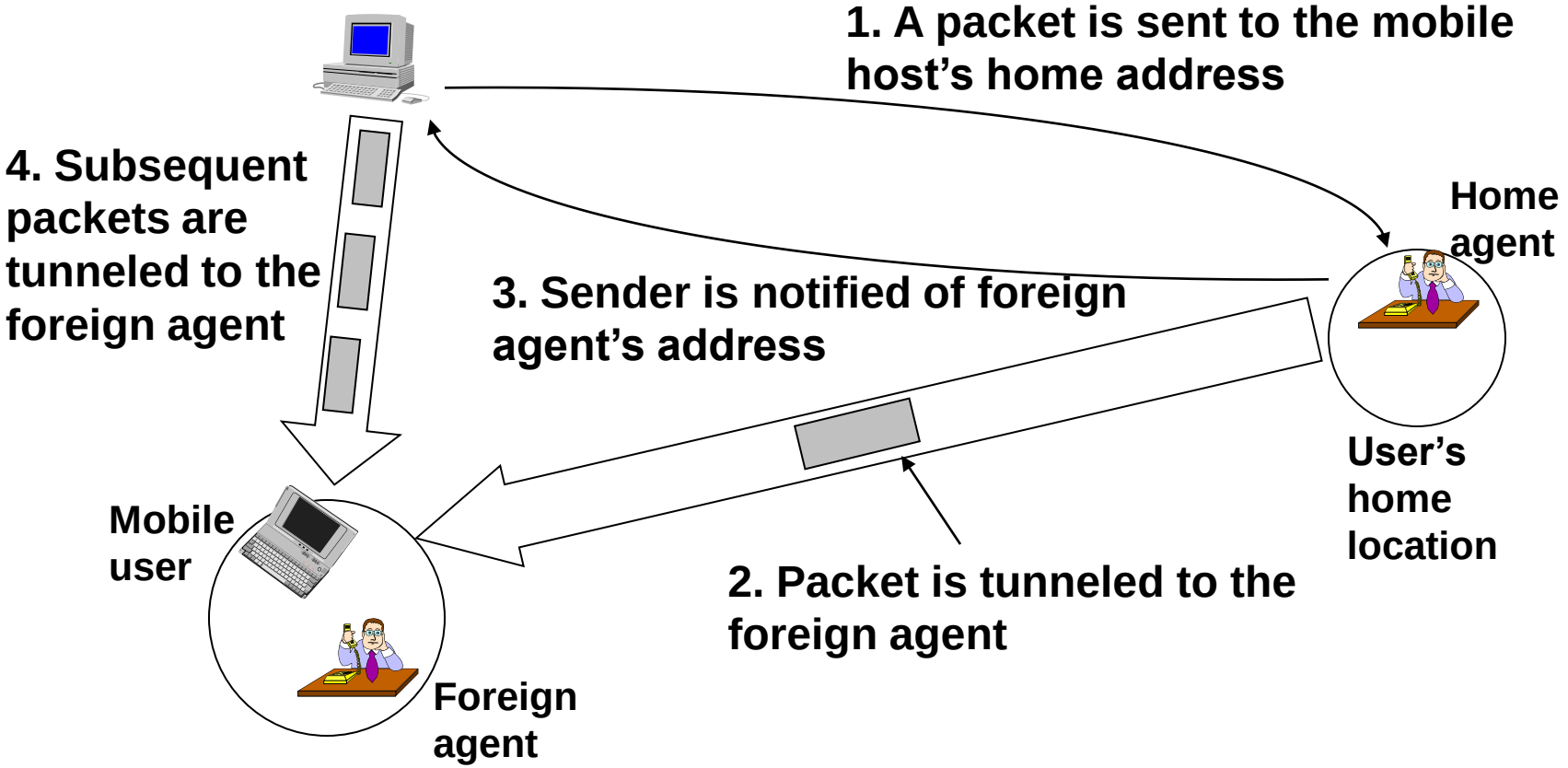


Hierarchical Routing (cont'd)

- A router has one entry, in its routing table, for each router in the same region, and also one representation entry for each of other regions.
- Example: For a subnet with 720 routers partitioned into 24 regions of 30 routers each, each router needs 53 entries (30 local + 23 remote).
- For a subnet with n routers, the optimal number of hierarchical levels is $\ln(n)$ and the number of entries per router is $e\ln(n)$.

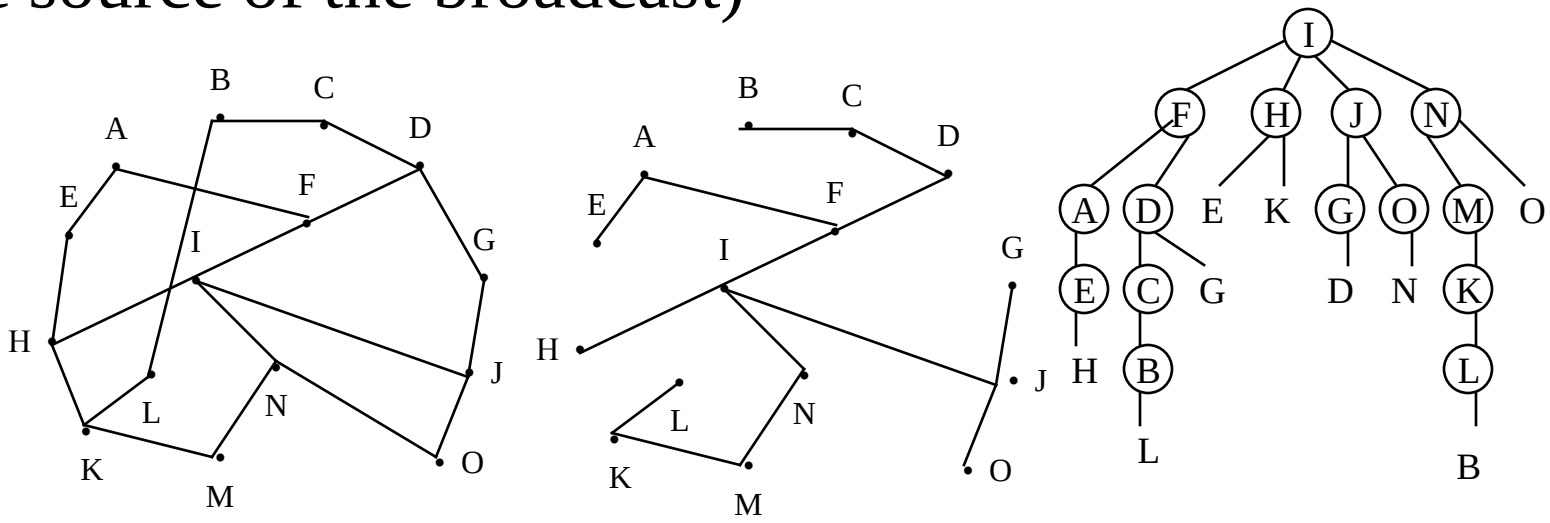
Routing for Mobile Hosts

- The mobile user first registers with the foreign agent, which then notifies the user's home agent.



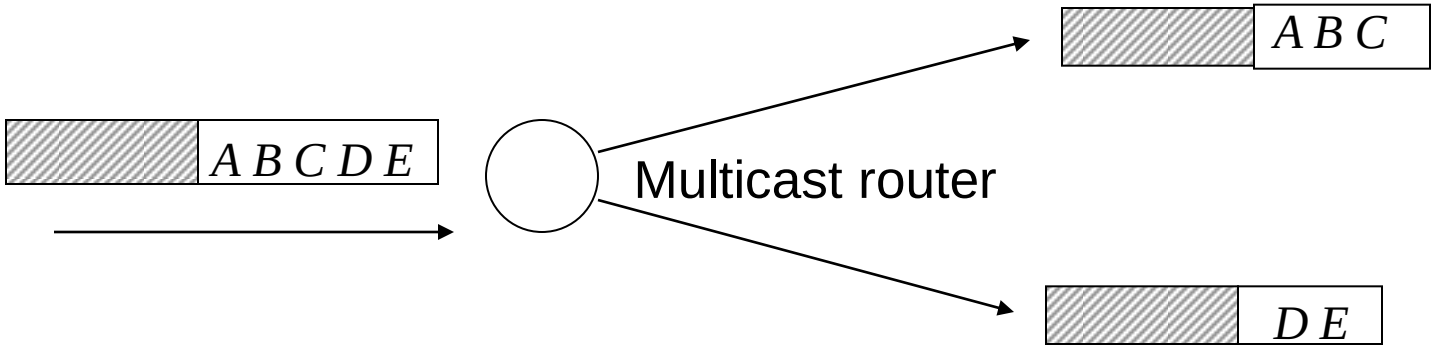
Broadcast Routing

- Possible methods: flooding, multi-destination routing, optimal sink tree, reverse path forwarding
- Reverse path forwarding: approximate the optimal sink tree (router checks to see if the packet arrived on the line that is normally used to send packets to the source of the broadcast)



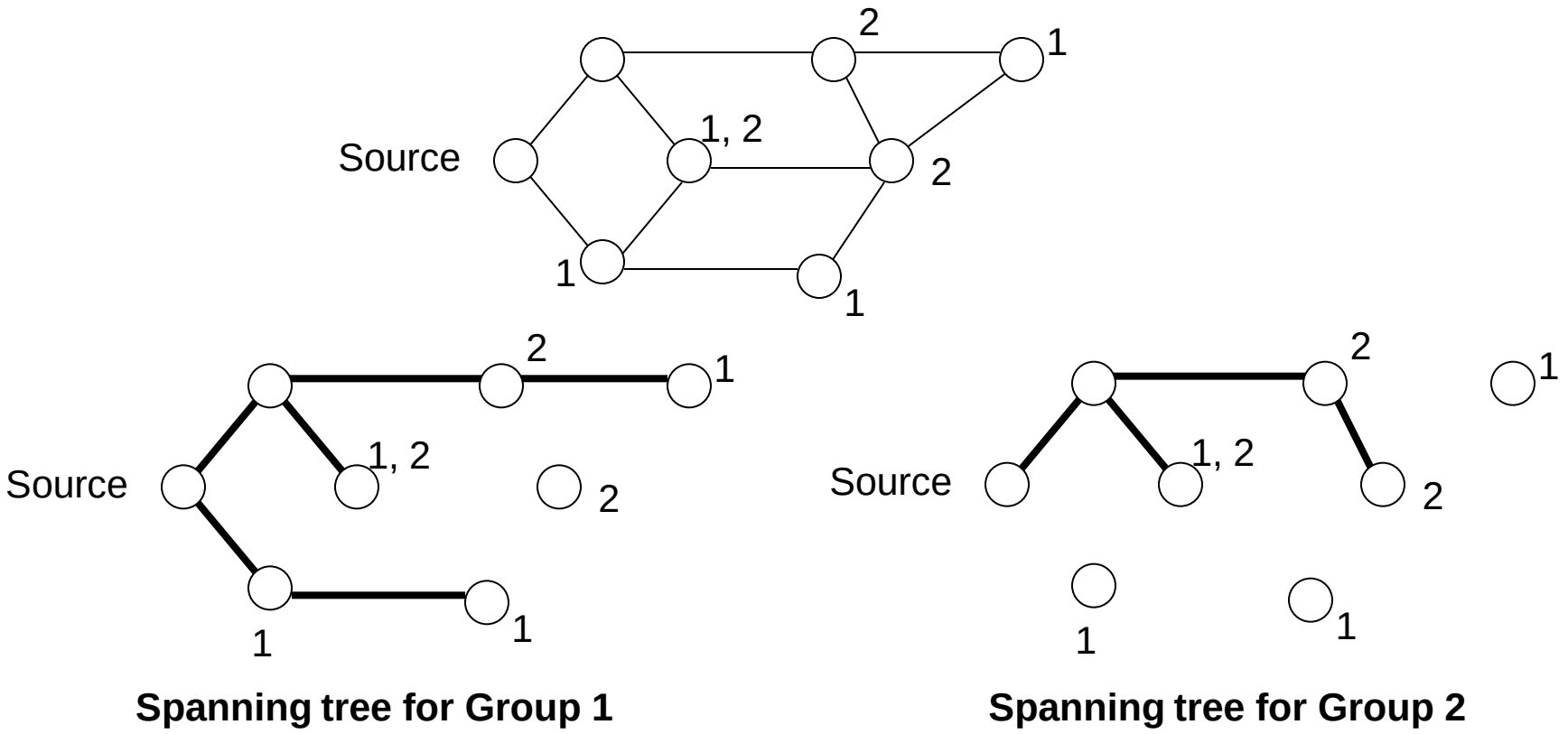
Multicast Routing

- Multicast: sending a message to a group of nodes
- Hosts may join or leave groups
- Routers must know which of their hosts belong to which groups, and inform other routers



- MBone has been operational since 1992 to multicast live audio and video on the Internet

Multicast Spanning Tree

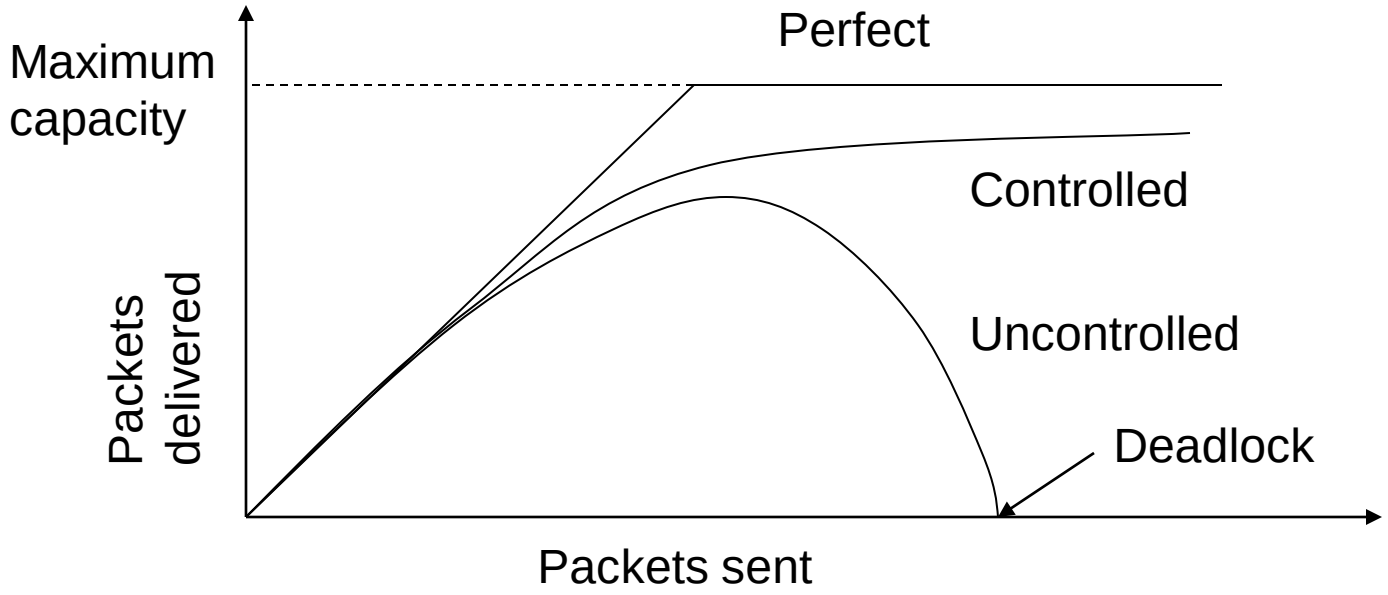


Spanning tree for Group 1

Spanning tree for Group 2

- Drawback: it scales poorly to large networks

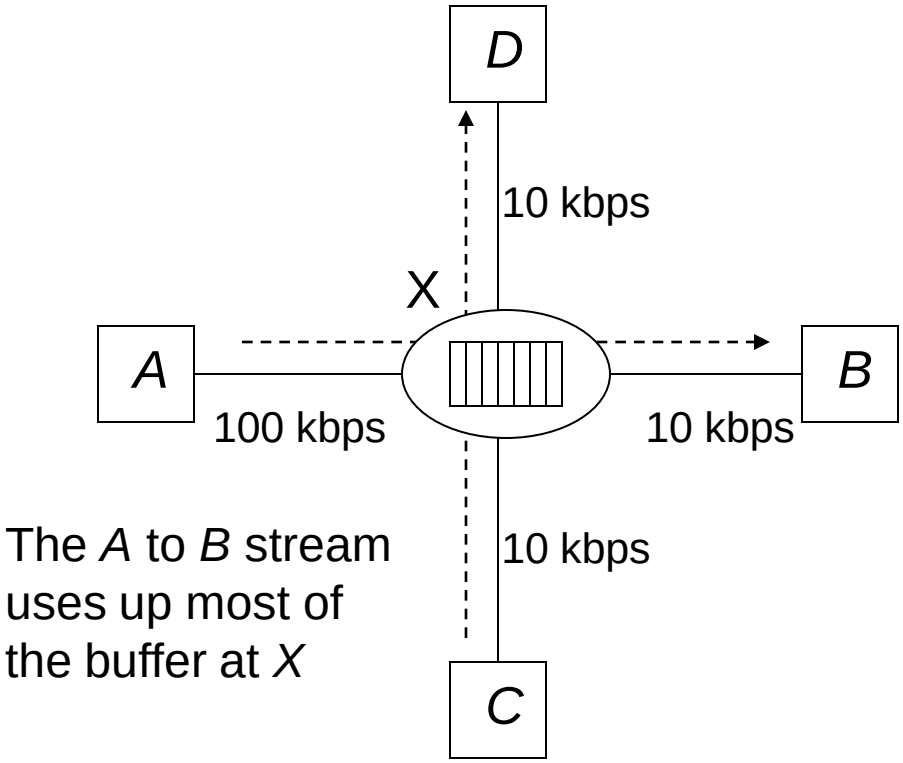
Congestion Control

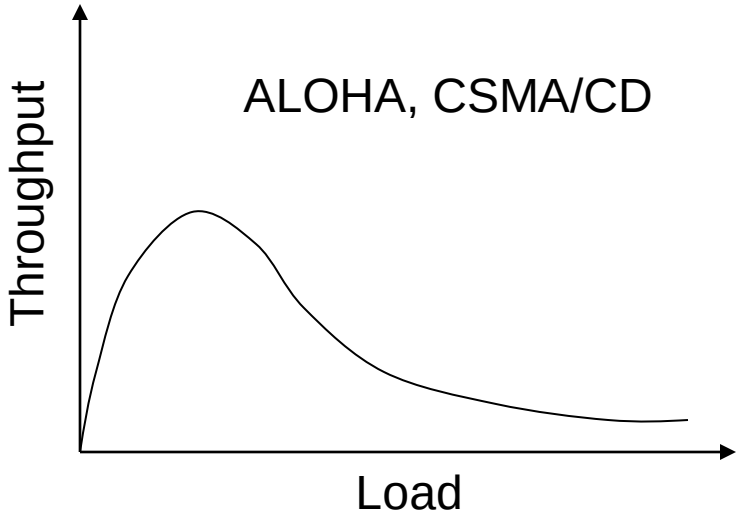


- Factors that cause congestion
  - insufficient buffer
  - slow CPU
  - low-bandwidth lines

} Need to upgrade both

Congestion Control (cont'd)

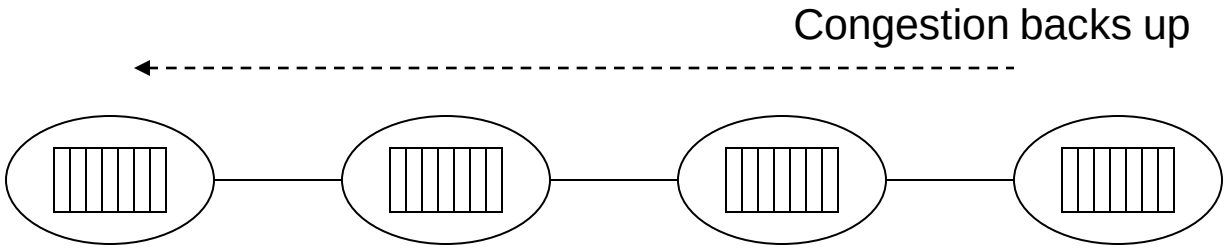




- Main reason: Uncontrolled sharing of resources (buffer, bandwidth, etc.)

Congestion Control (cont'd)

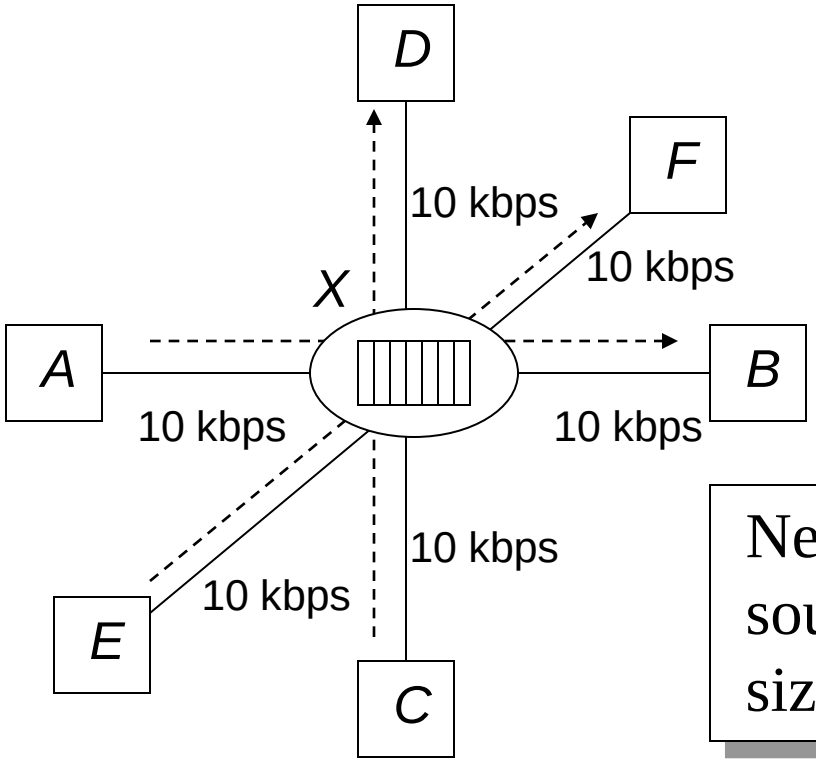
- Congestion tends to feed upon itself



- Congestion control
  - make sure the network is able to carry the offered traffic
- Flow control
  - make sure the sender does not overload the receiver in an point-to-point (or end-to-end) connection

Congestion Control (cont'd)

- The flow control (sliding window protocol) at the data link layer does not prevent congestion at the network layer



Need a way to slow down the source, i.e., change the window size dynamically

Congestion Control Principles

- Preventive control: take actions way before congestion ever happens
  - action at source
  - action at destination
- Corrective control: detect congestion via feedback and take corrective actions
  - 1. Detect 2. Inform source 3. Action
  - Explicit feedback
  - Implicit feedback
- Action: increase capacity, or decrease load

Policies That Affect Congestion

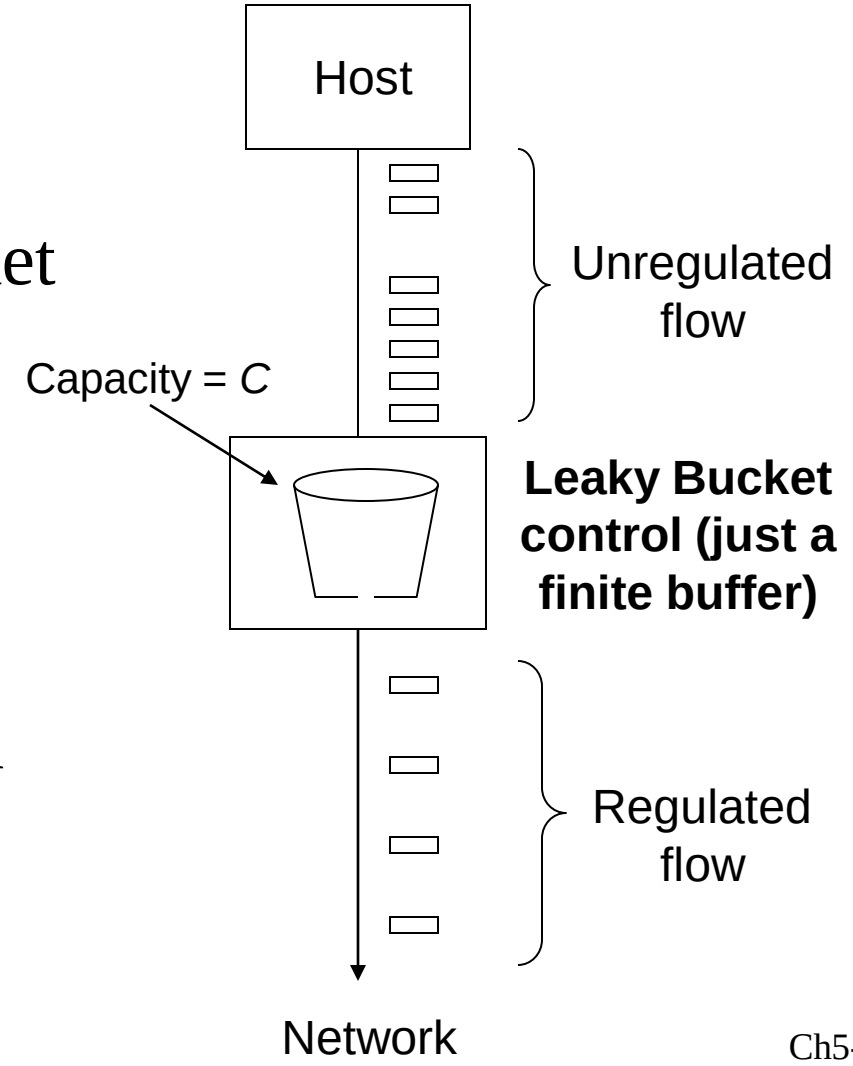
LAYER	POLICIES
Transport Layer	• Retransmission policy • Out-of-order caching policy • Acknowledgment policy • Flow control policy • Timeout interval
Network Layer	• VC versus DG routing • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data Link Layer	• Retransmission policy • Out-of-order caching policy • Acknowledgment policy • Flow control policy

Traffic Shaping

- A preventive control scheme
- Force the source to transmit packets in a more predictable way (different from sliding window control)
- Source and the network agree on a traffic pattern during VC setup
- Algorithms
  - Leaky Bucket Algorithm
  - Token Bucket Algorithm

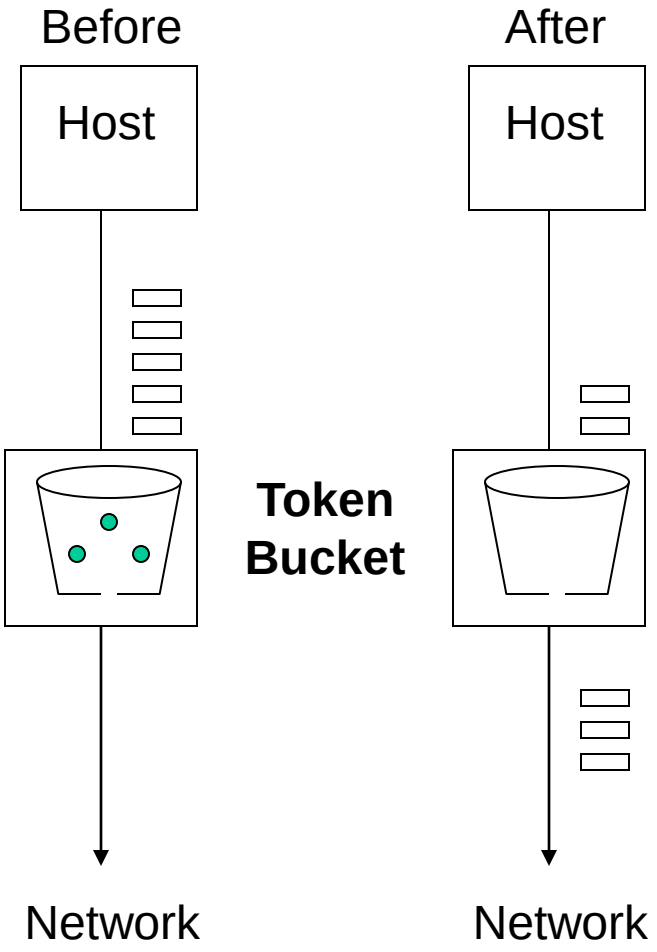
The Leaky Bucket Algorithm

- Each host is connected to a leaky bucket interface
- The bucket allows one packet to pass every ΔT sec
- If a packet arrives and the bucket is full, the packet is discarded
- The output rate is very rigid

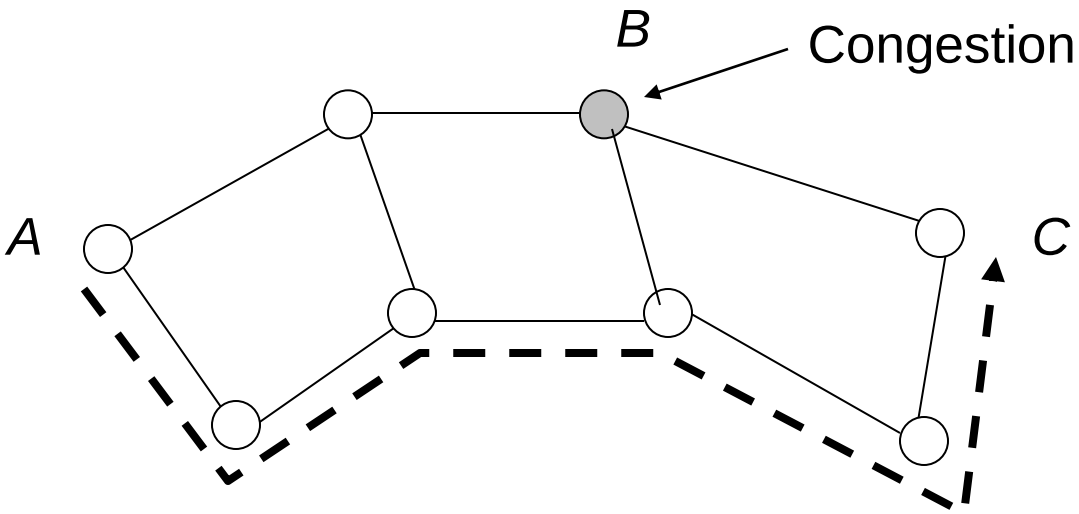


The Token Bucket Algorithm

- A token is generated every ΔT sec
- The bucket can hold at most n tokens
- Each packet must capture a token before it can be transmitted
- Host negotiates with the network on
  - n, ΔT, max packet size, max transmission rate

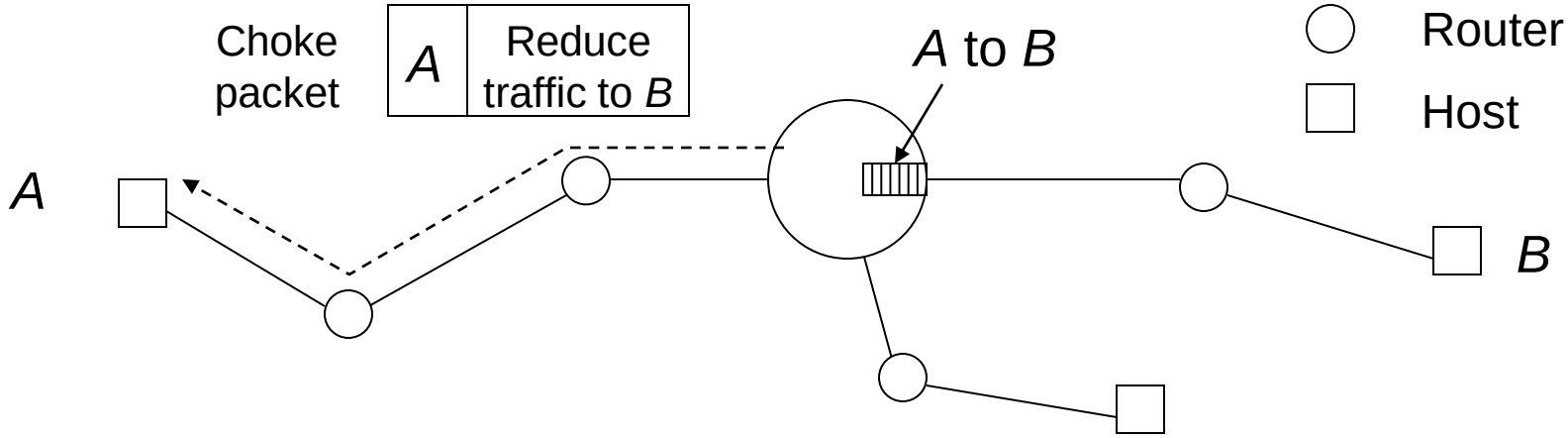


Admission Control



- When congestion occurs, allow no new VC (*A* to *B*) setup, or route new VC (*A* to *C*) around congested area
- Works on VC only

Choke Packets



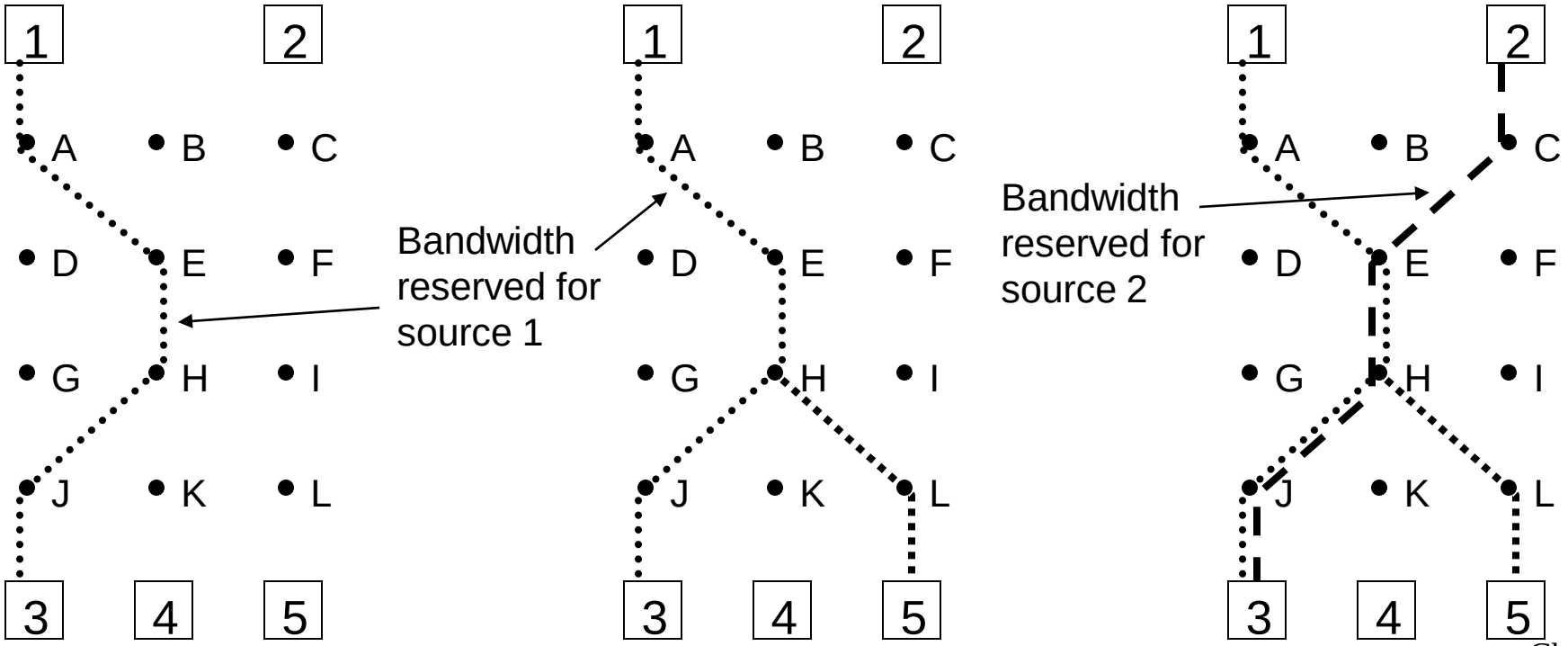
- A choke packet is sent to the source when the output line is congested
- The source reduces its traffic to *B* by certain percentage (e.g., 50 % each time)
- A variation is to have the choke packet take effect at every hop it passes through

Load Shedding

- A router drops packets when it is congested
- Which packets to drop depends on applications, e.g.,
  - for file transfer: dropping young packet is better
  - for real time application: dropping old packet is better
- Applications mark their packets to different priority classes
  - low priority to be dropped first at congestion
  - need policy to enforce this

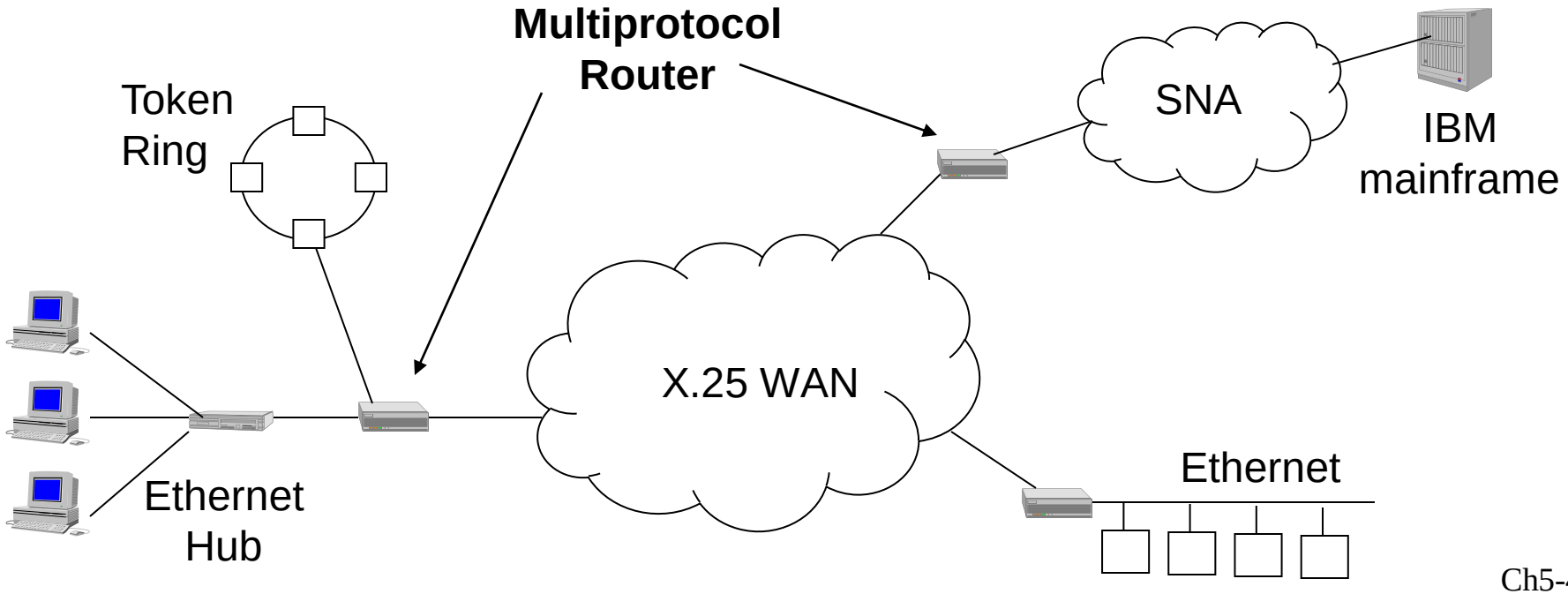
Congestion Control for Multicasting

- RSVP - Resource reSerVation Protocol
- The receiver initiates the bandwidth reservation before receiving traffic



Internetworking

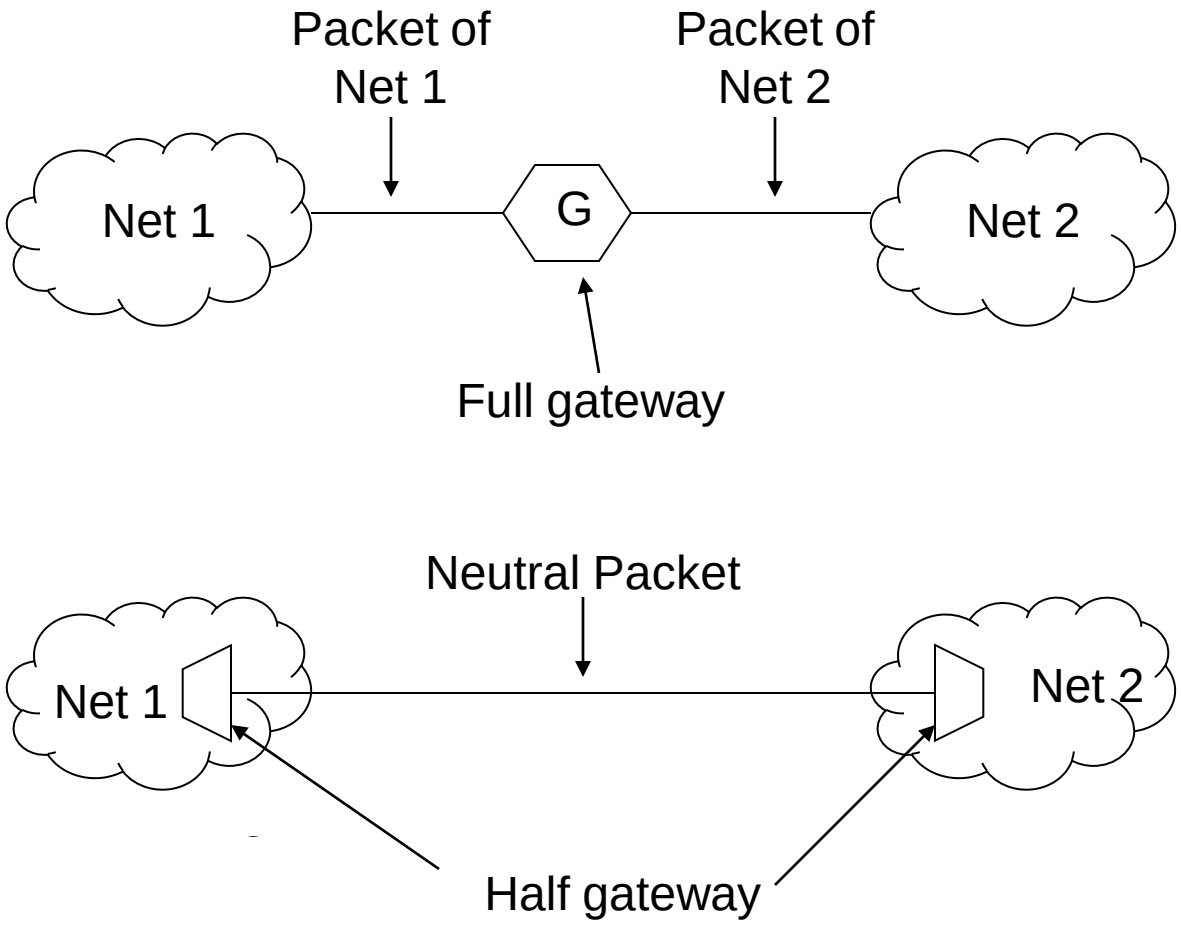
- Different networks exist today: TCP/IP, SNA, DECnet, SPX/IPX, AppleTalk, ATM, Wireless
- Networks differ at protocols, VC/DG, addressing, packet size, QoS, etc.



Interconnecting Devices

- Layer 1: repeater ---- copy bit by bit
- Layer 2: bridge (hub, Ethernet switch)
  - store and forward frames
  - interconnect multiple LANs
- Layer 3: Multiprotocol routers
  - store and forward packets
  - interconnect different types of networks (IP, IPX,..)
- Layer 4: Transport gateway
  - e.g., conversion between TCP and ATM connections
- Layer 7: Application gateway
  - e.g., conversion between different email packages

Full Gateway and Half Gateway



Internetworking Scenarios

- Local area LAN interconnection
  - Bridge, switch, router
- Wide area LAN interconnection (i.e., LAN-WAN-LAN)
  - Modem, leased-line, ISDN, X.25, Frame Relay, SMDS (Switched Multimegabit Data Services), ATM

Concatenated VCs

Fig 5-36

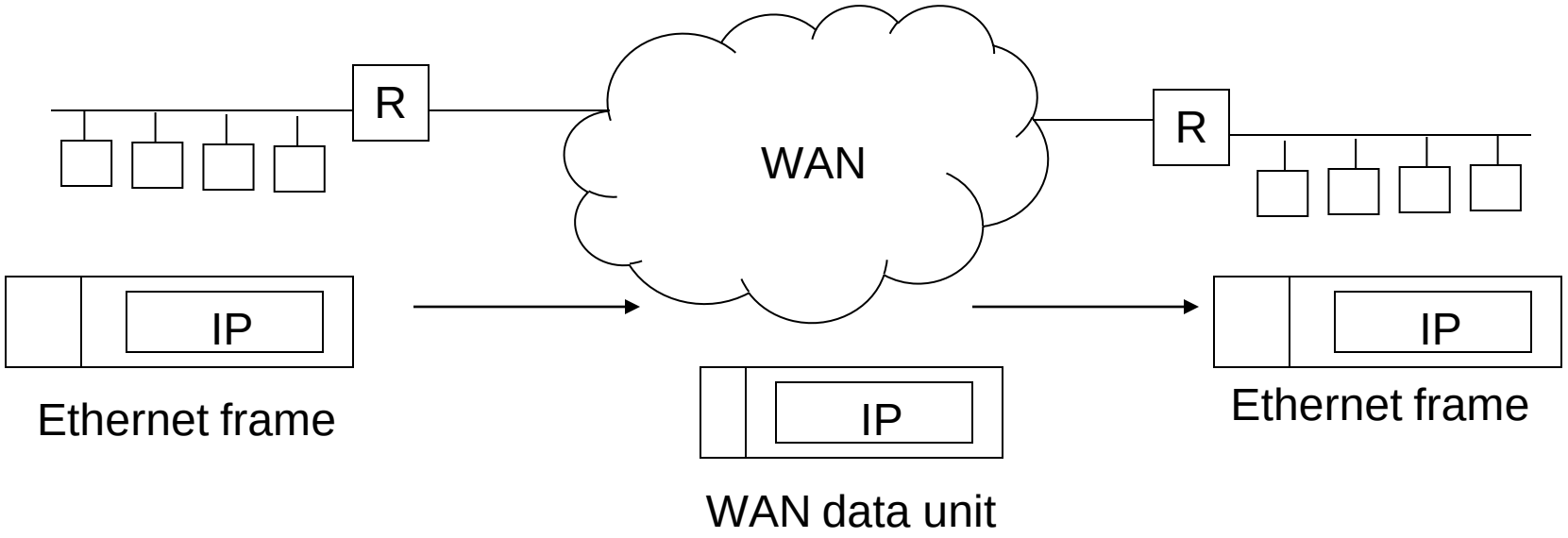
- Packets basically follow the same route
- This works best if all the networks have roughly the same properties

Connectionless Internetworking

Fig 5-37

- Packets may be routed over multiple routes

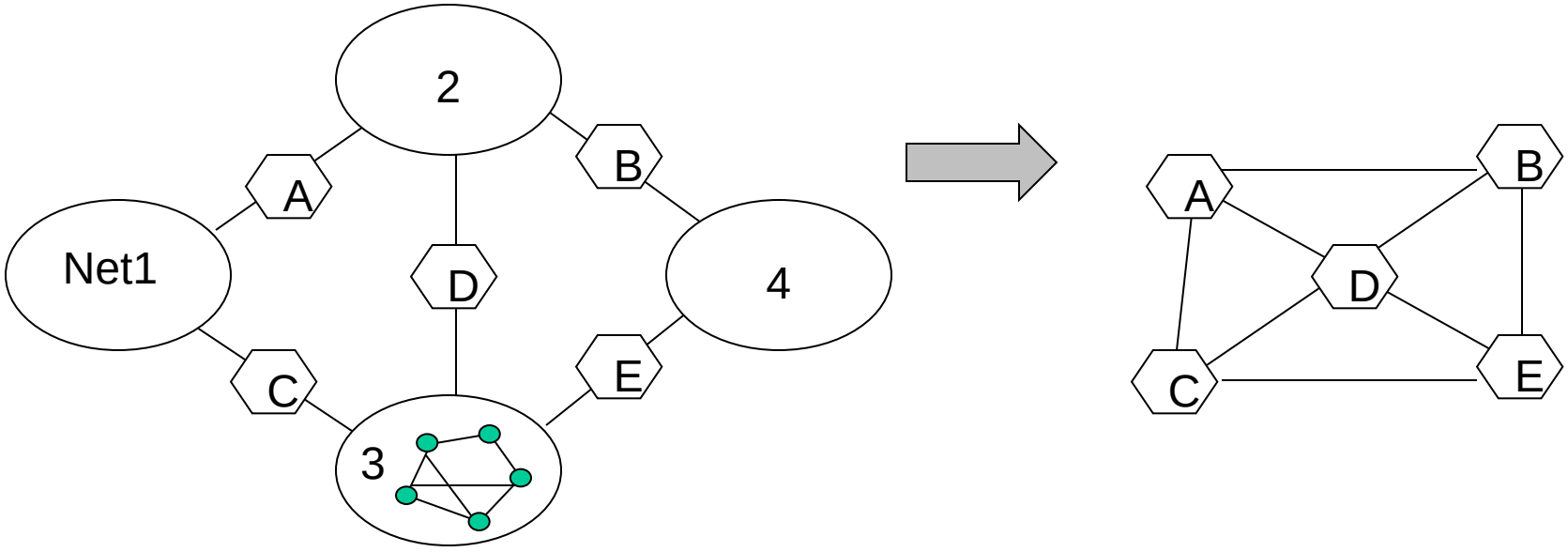
Tunneling



- When source and destination networks are of the same type

Internetwork Routing

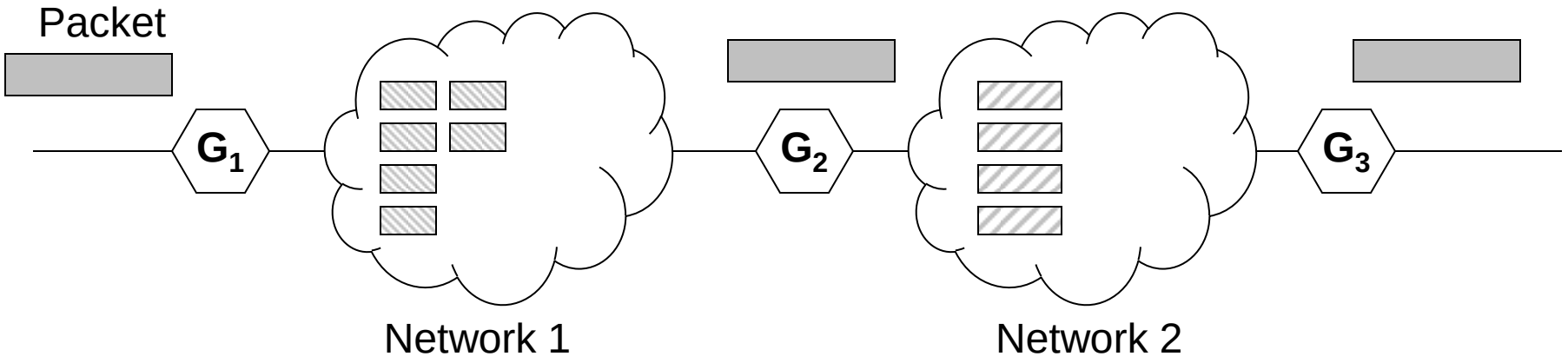
- An Exterior Gateway Protocol (EGP) is used for routing between the networks
- An Interior Gateway Protocol (IGP) is used for routing within each network



Fragmentation

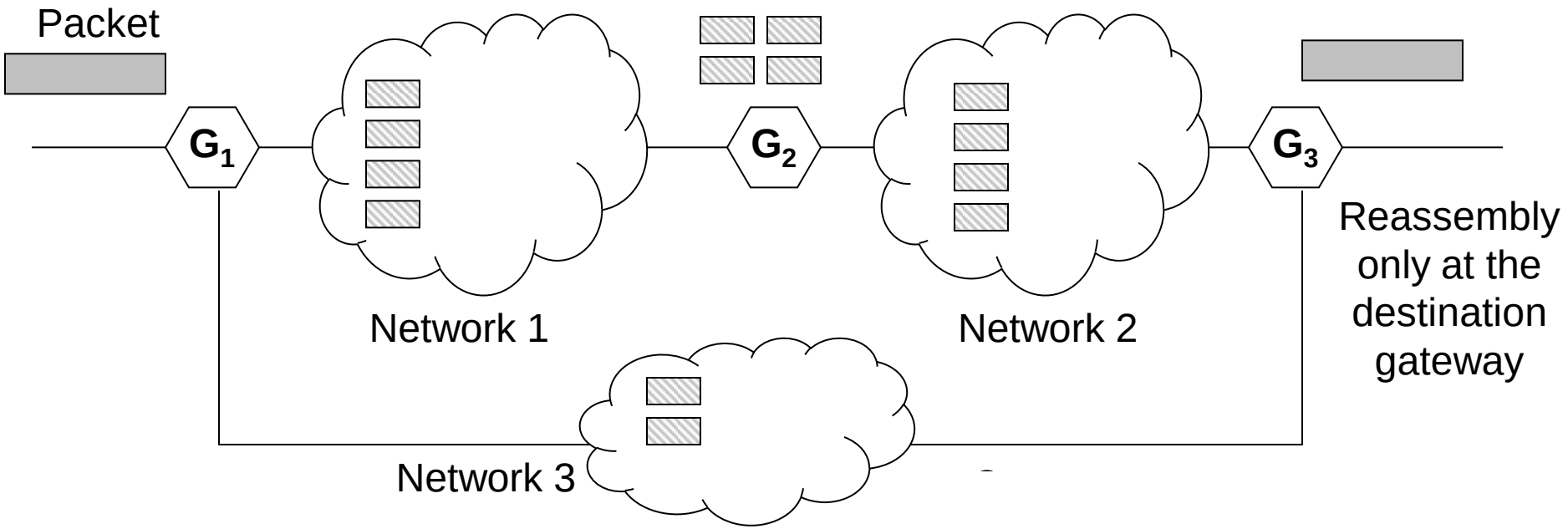
- Different networks may use different packet size (e.g., 48 bytes in ATM and 65,515 in IP) because
  - line efficiency
  - error rate
  - buffer size
  - minimize delay for priority traffic
- Two possible approaches
  - transparent fragmentation
  - nontransparent fragmentation

Transparent Fragmentation



- Reassembly at each gateway
- For example, IP packet across an ATM network
- Pros: simple, transparent
- Cons: frequent fragmentation/reassembly

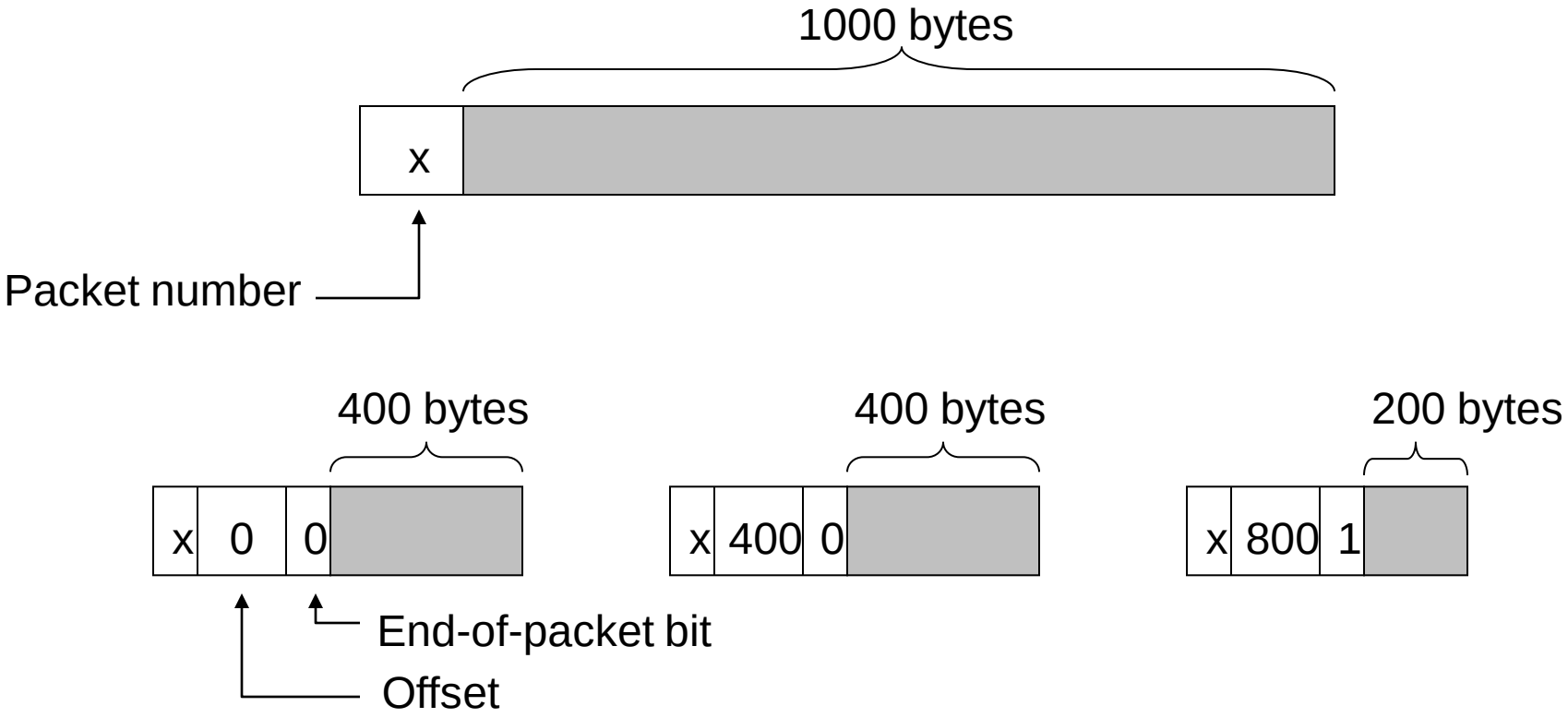
Nontransparent Fragmentation



- For example, the packet is broken into six fragments, four routed via Network 1, and two via Network 3
- Pros: less fragmentation/reassembly, multiple routes can be used
- Cons: large header overhead

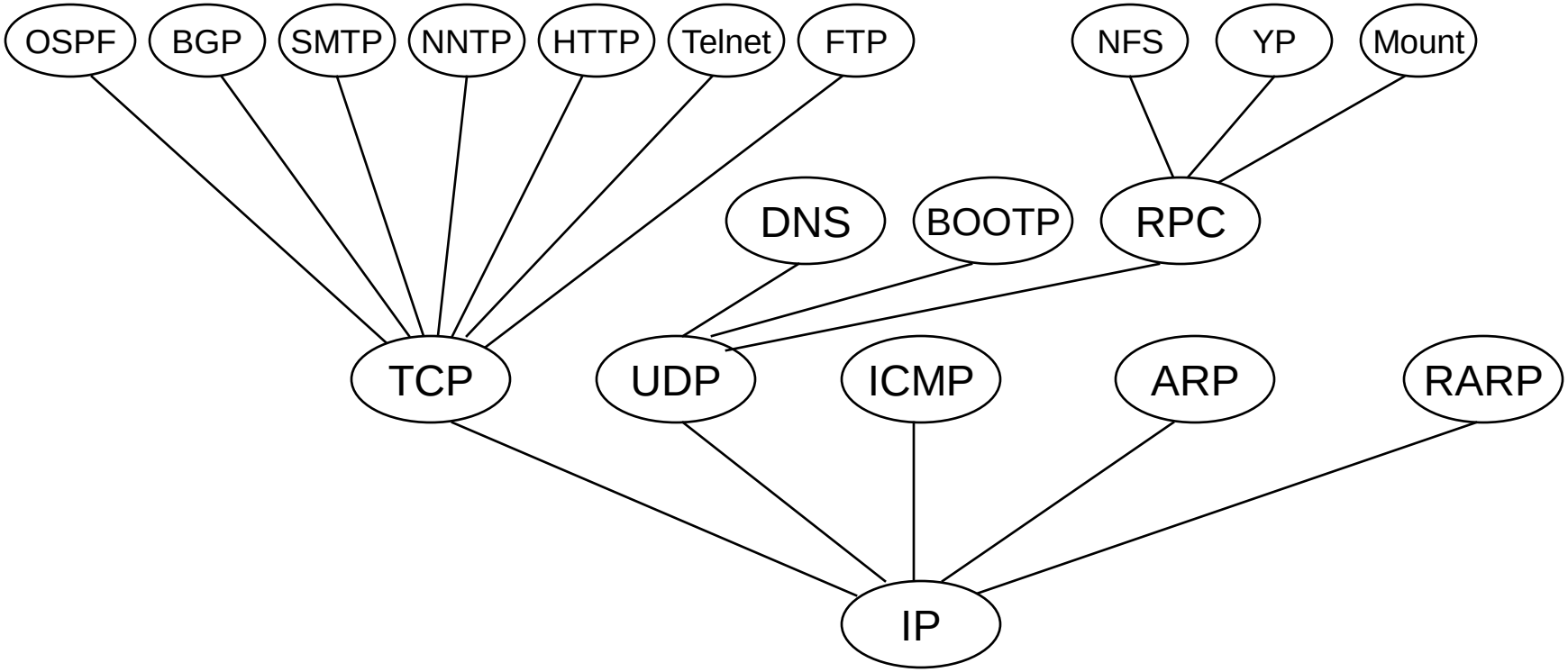
Fragment Numbering

- Suppose a packet is broken into three fragments

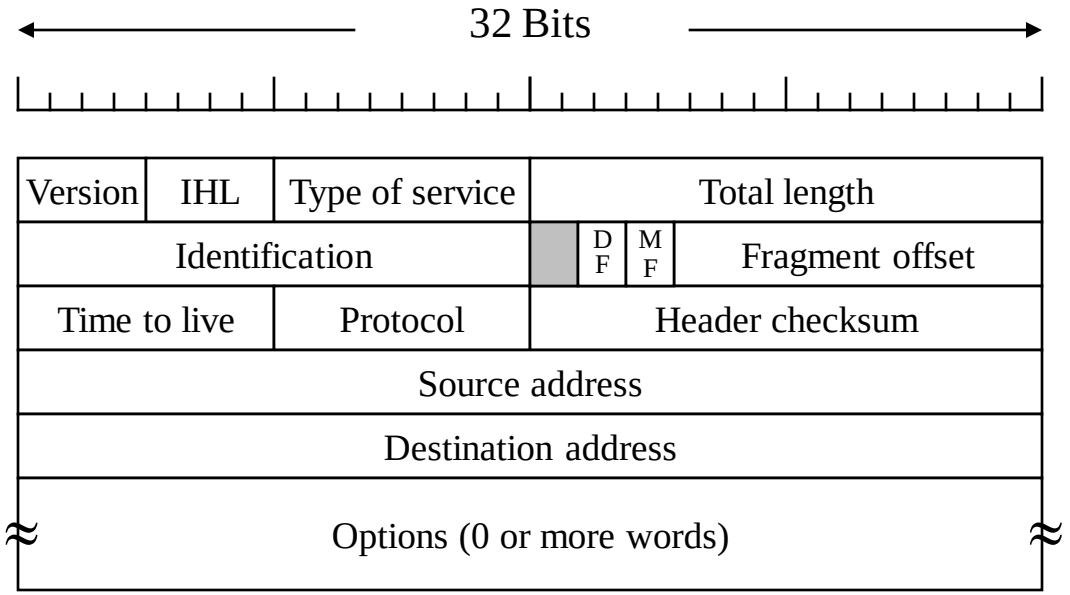


- Standard requires that every IP network must accept 576-byte fragments

Internet Protocol Hierarchy



IP Header



IHL: Header length (between 20 and 60)
Type of service: priority, … (ignored today)
DF: Don't fragment
MF: More fragment (i.e., the end-of-packet bit, set only in the last fragment)

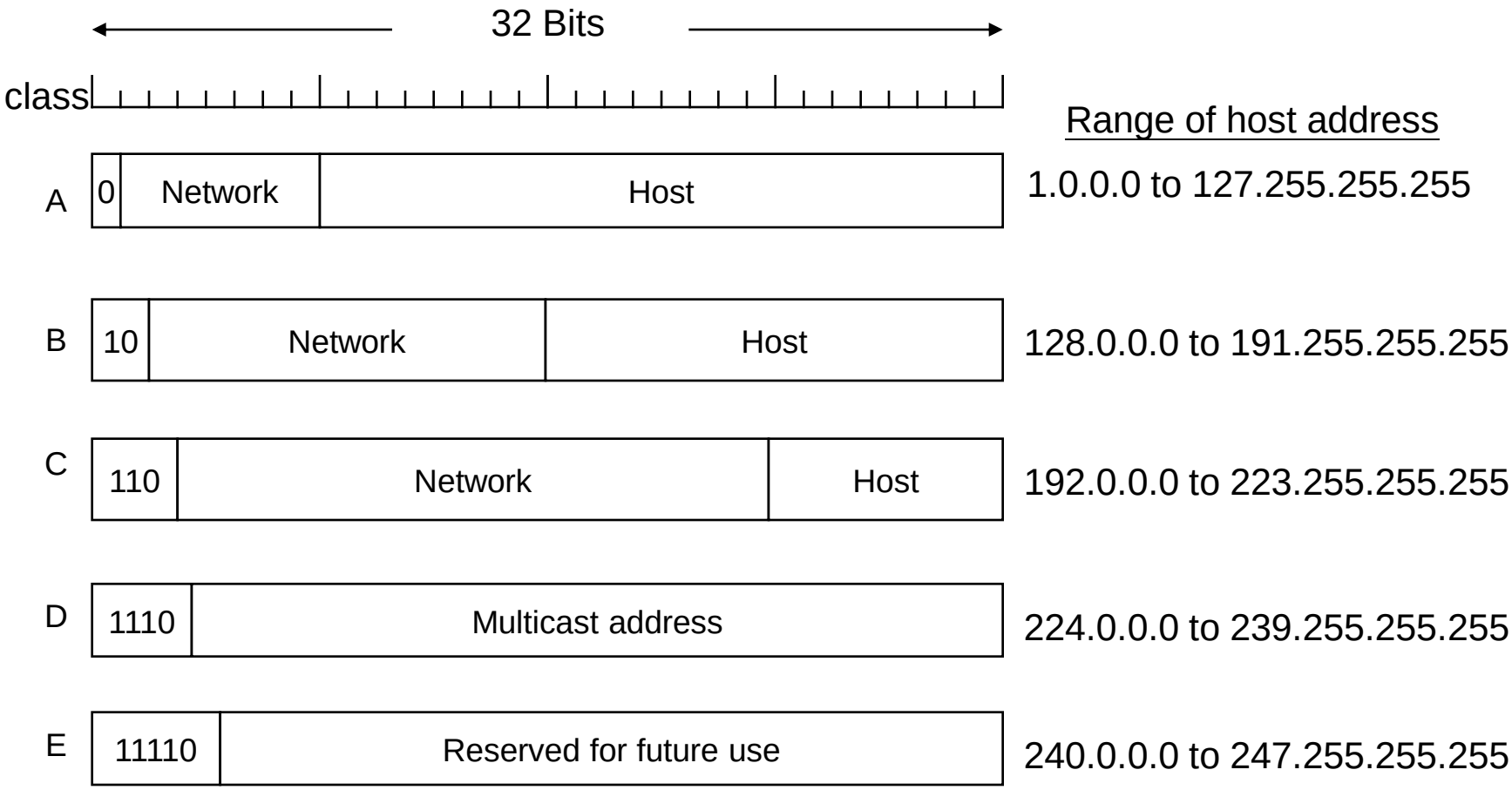
Fragment offset: in multiples of 8 bytes (i.e., offset = 2 means 16 bytes)
Time to live: hop count, decremented by each router
Protocol: TCP, UDP

IP Header Options

- Option field cannot be longer than 40 bytes

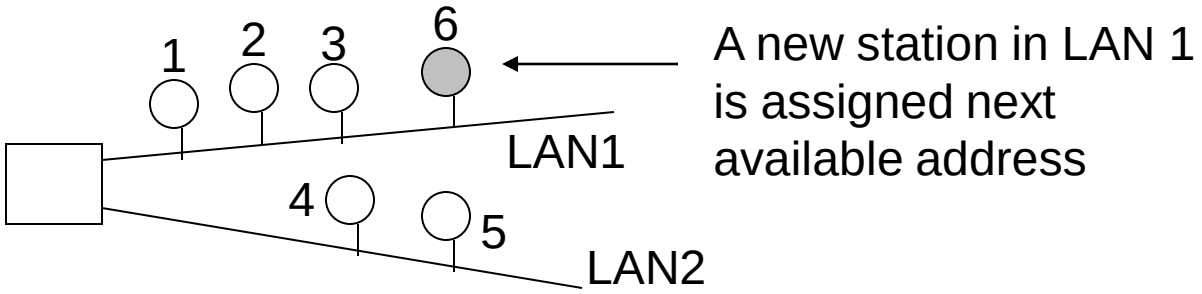
Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Asks each router to append its IP address
Timestamp	Asks each router to append its address and timestamp

IP Address

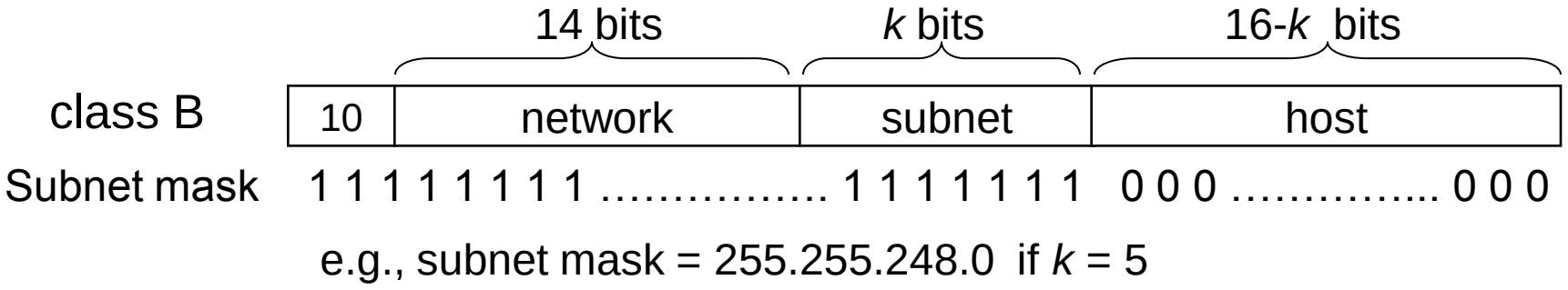

32 Bits
class
Range of host address
A
0
Network
Host
1.0.0.0 to 127.255.255.255
B
10
Network
Host
128.0.0.0 to 191.255.255.255
C
110
Network
Host
192.0.0.0 to 223.255.255.255
D
1110
Multicast address
224.0.0.0 to 239.255.255.255
E
11110
Reserved for future use
240.0.0.0 to 247.255.255.255

IP Subnetting

- A class B network can hold up to 65534 hosts
- Such a large flat address space is hard to manage



- So divide the host field into subnet and host fields



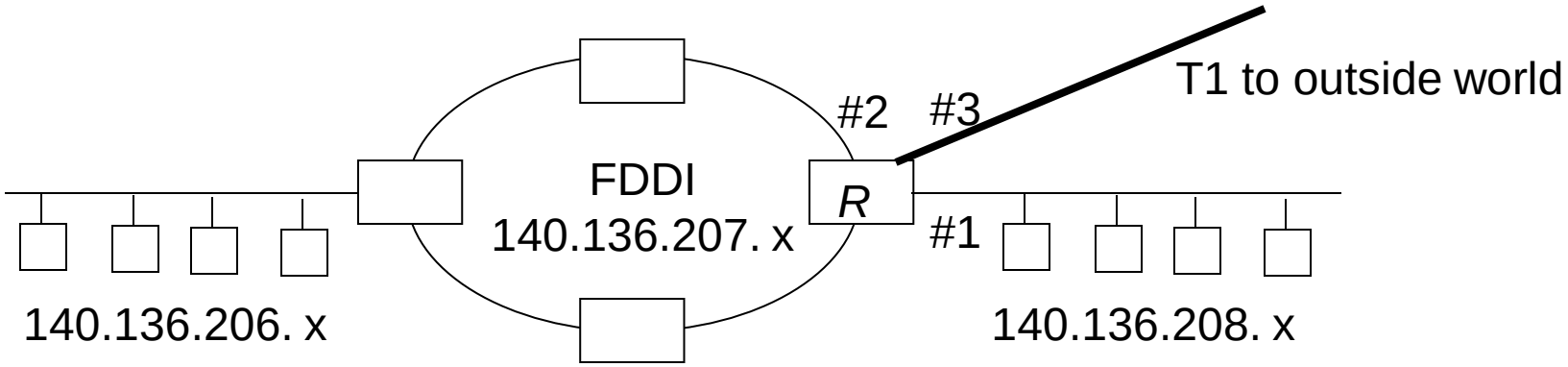
IP Subnet Routing

	Destination IP	= 140.136.208.129	=10001100 10001000 11010000 10000000
AND	Subnet mask	= 255.255.255. 0	=11111111 11111111 11111111 00000000
	Subnet address	= 140.136.208. 0	=10001100 10001000 11010000 00000000

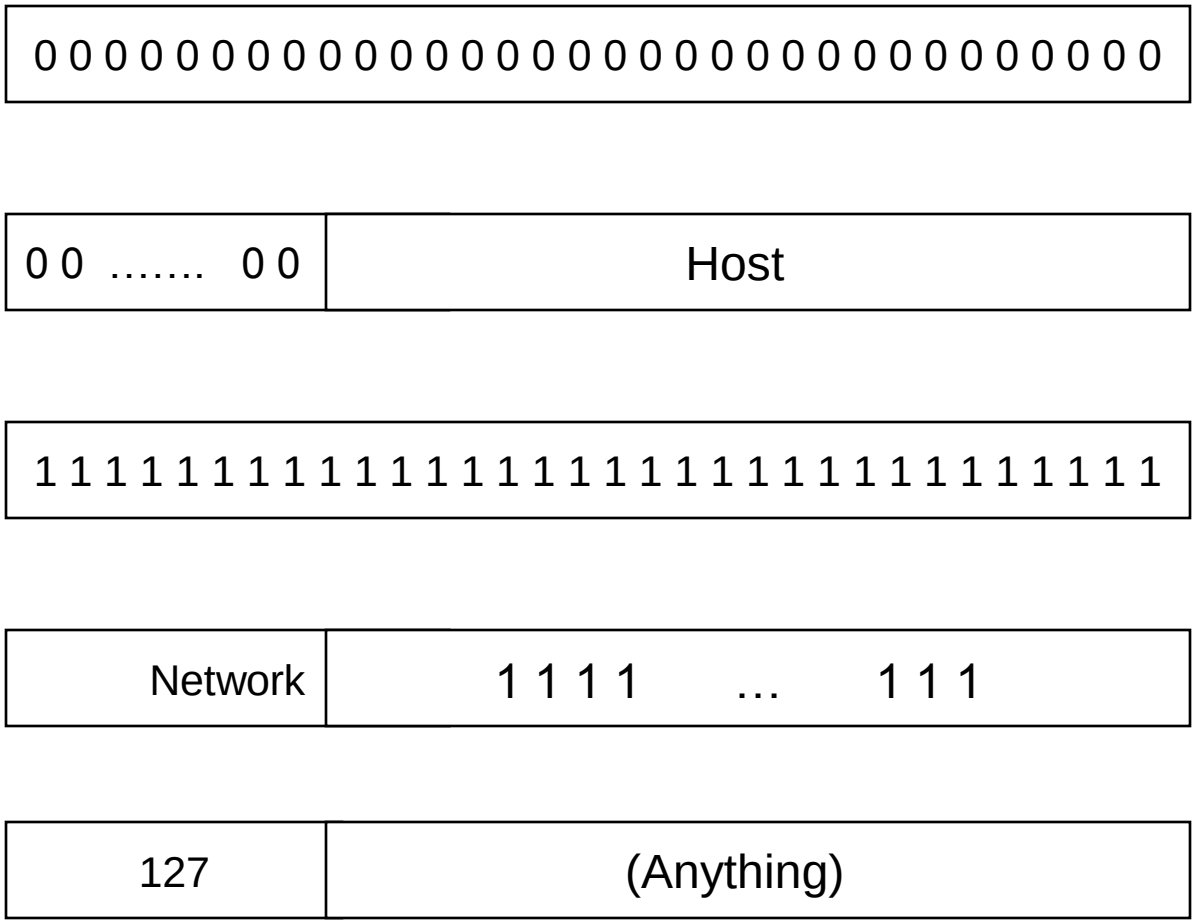
Use the subnet address to look for the entry with longest prefix match

Routing table at router R

Subnet address	Output port
140.136.208	#1 (Ethernet)
140.136	#2 (FDDI)
others	#3 (T1)



Special IP addresses



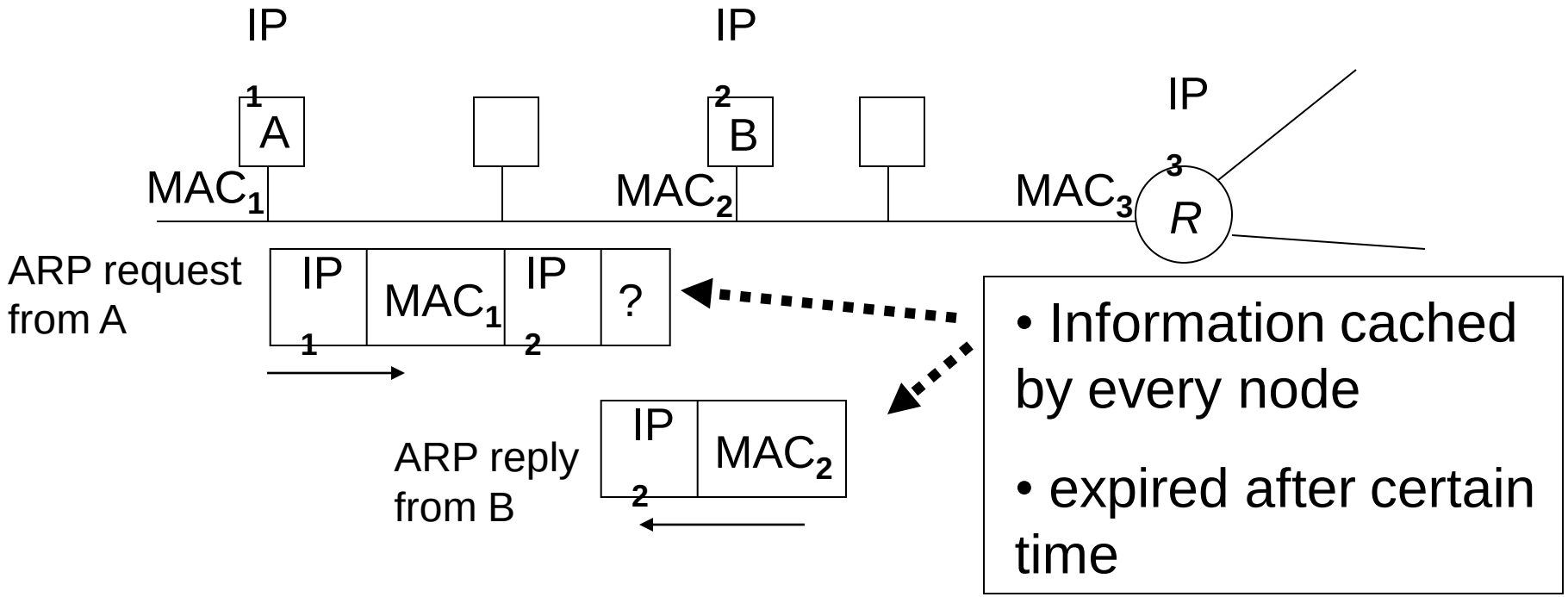
ICMP (Internet Control Message Protocol)

- To test the network or to report events

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 request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

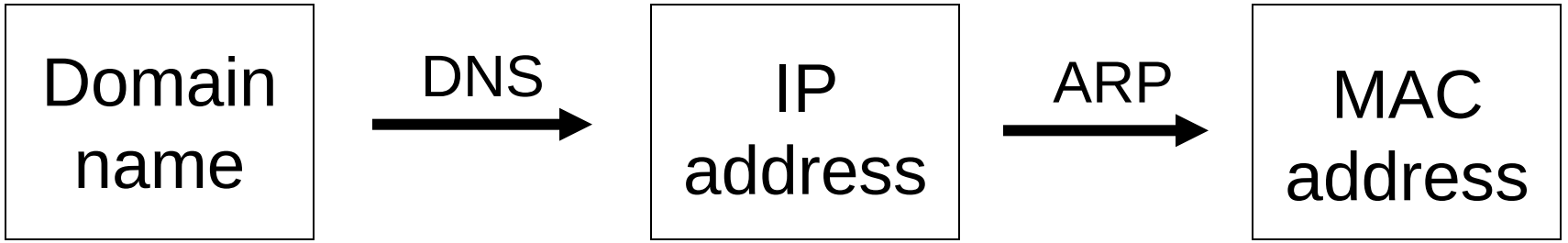
ARP (Address Resolution Protocol)

- To resolve the mapping of IP and MAC address
- ARP runs on every machine, including PC



- If nobody responds, send the packet to a default node, i.e., the router *R* (called gateway in Win95)

Address Resolutions



optimal.im.ntu.edu.tw

140.112.106.60

FF.3E.29.1A.53.2C

DNS = Domain Name System

RARP (Reverse ARP)

- RARP
  - Mapping of MAC address to IP address
  - For example, can be used by a diskless station to obtain an IP address from a server after booting up
  - The RARP server must be on the same LAN as the diskless station
- BOOTP and DHCP (Dynamic Host Control Protocol) are two protocols that allow the server to be on a remote network
  - Can also provide additional information such as subnet mask, default router, where to download OS, etc.

OSPF (Open Shortest Path First)

- Internet is made up of many AS (Autonomous System), with each AS operated by a different organization
- OSPF is the commonly-used IGP (interior gateway protocol) routing algorithm within an AS
  - Based on link state routing
  - A serial connection between two routers is represented by a pair of arcs, one in each direction, with possibly different weight
  - A serial connection can be a point-to-point line, a LAN, or a WAN

OSPF (cont'd)

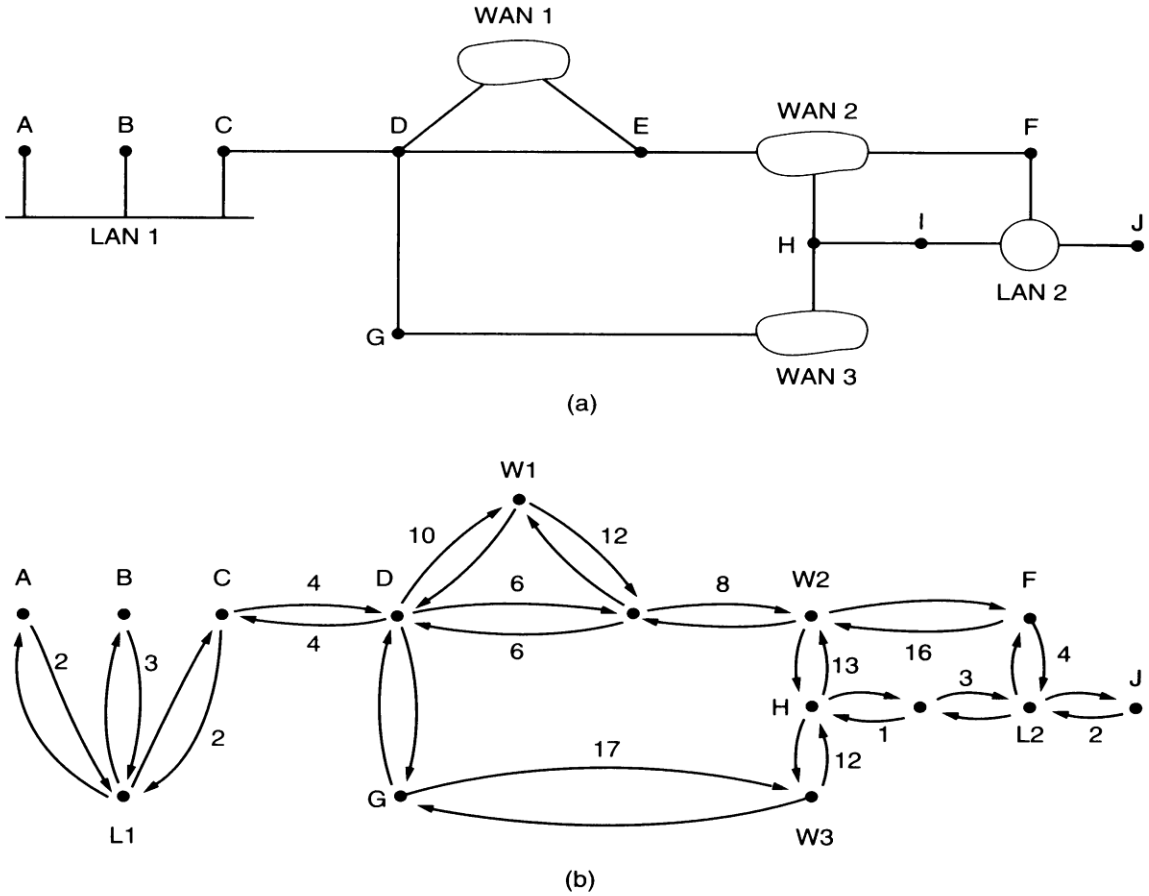


Fig. 5-52. (a) An autonomous system. (b) A graph representation of (a).

OSPF (cont'd)

- Each AS may be divided into areas
  - There exists a backbone area that connects directly to all the other areas in the AS
- Three types of routes
  - Intra-area: link state shortest path routing
  - Inter-area: always go through the backbone area
  - Inter-AS: use BGP (Border Gateway Protocol), which is a type of EGP (Exterior Gateway Protocol)

OSPF (cont'd)

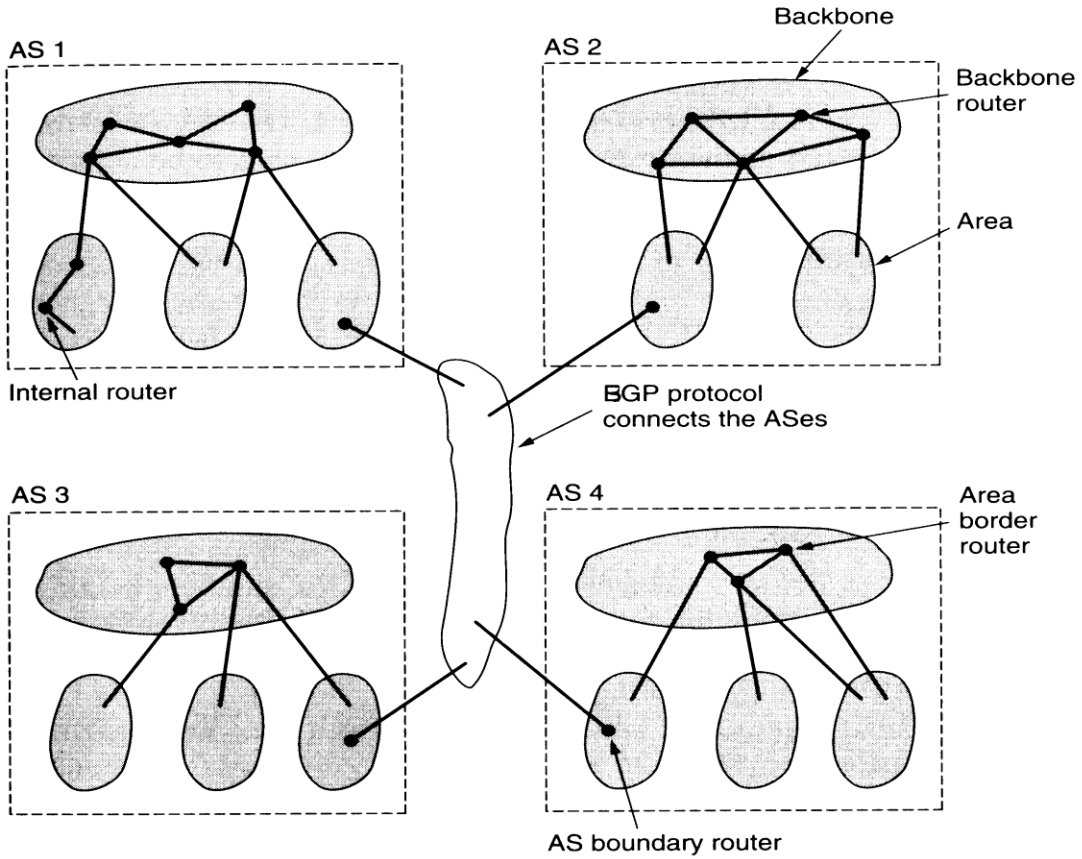
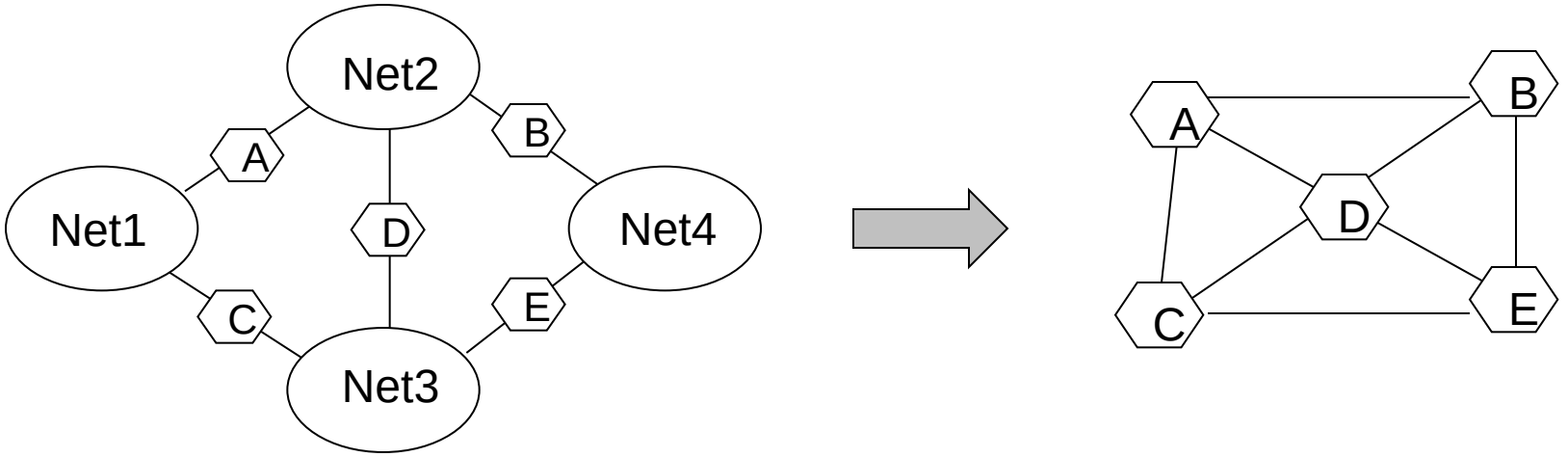


Fig. 5-53. The relation between ASes, backbones, and areas in OSPF.

BGP (Border Gateway Protocol)

- BGP is used for routing between Ases
  - BGP is fundamentally a distance vector protocol, but
    - each node records the cost and the exact path for each destination
    - exchanges the above information with its neighbors periodically
    - routing policies concern with politics a great deal. Any route violating policies will not be chosen

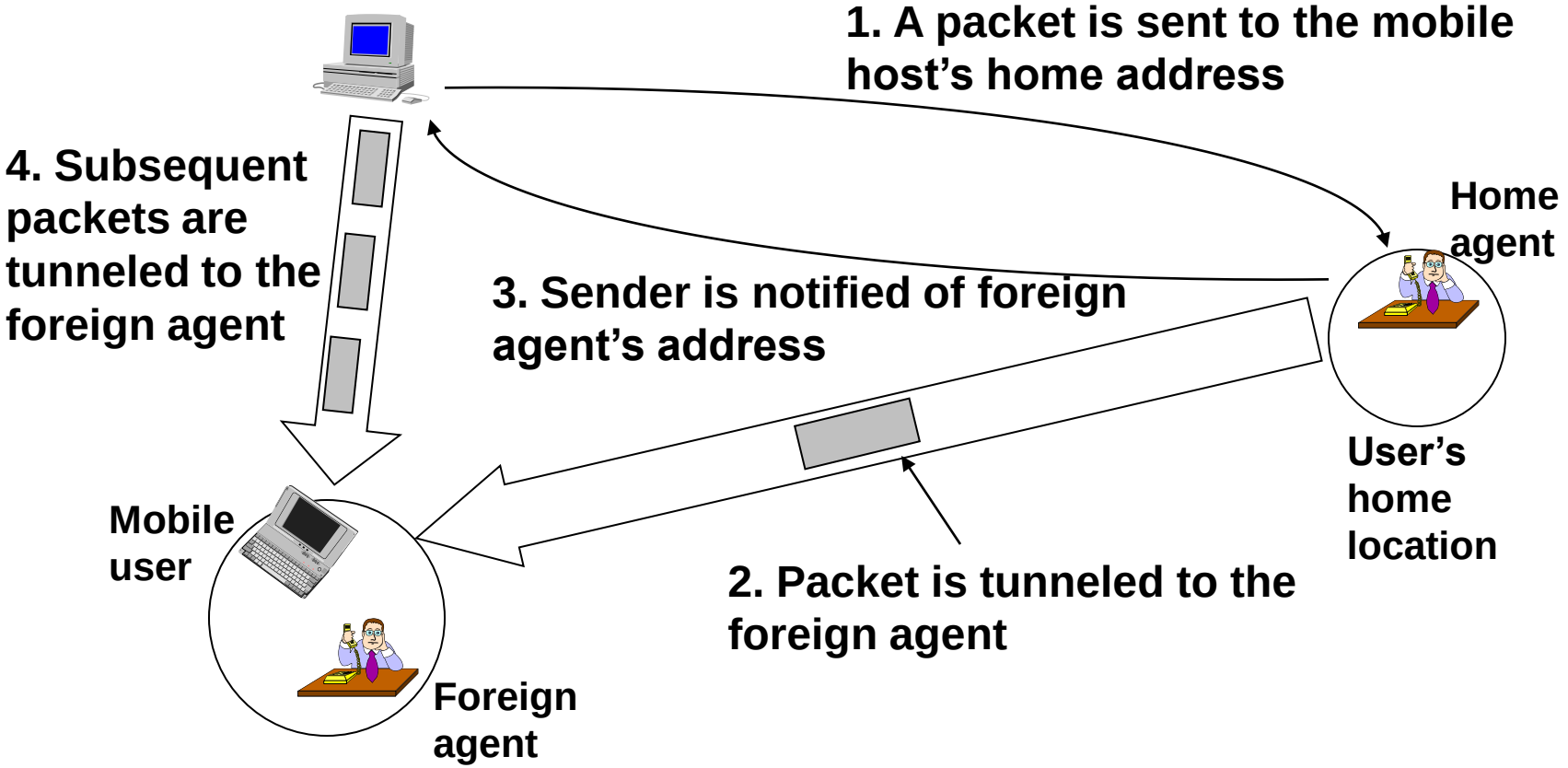


IGMP (Internet Group Management Protocol)

- Group addresses for multicasting
- Permanent groups:
  - 224.0.0.1 all systems on a LAN
  - 224.0.0.2 all routers on a LAN
  - 224.0.0.5 all OSPF routers on a LAN
  - 224.0.0.6 all designated OSPF routers on a LAN
- Temporary groups:
  - IGMP query: each multicast router multicasts to hosts on its LAN to ask them the groups they belong to
  - IGMP response: each host responds with the class D addresses it is interested in
- Each multicast router constructs a pruned spanning tree per group, using a modified distance vector protocol

Mobile IP

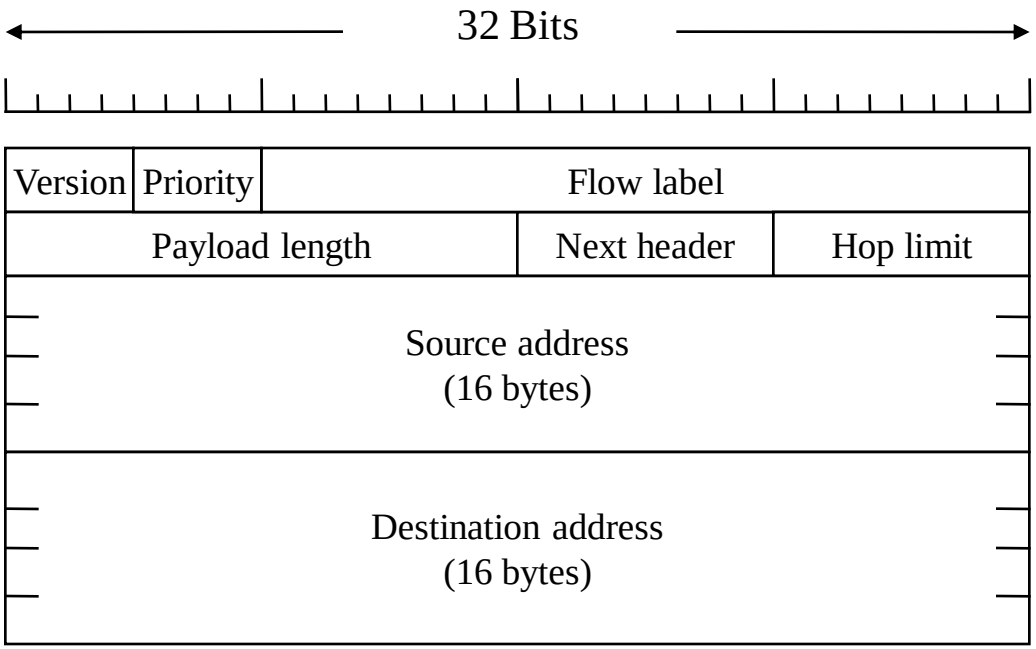
- To use the same IP address no matter where you are



IPv6

- Objectives
  - more IP addresses, reduce routing table size, better security, Type of Service support, faster processing, etc.
- IPv6 improvements
  - 16 bytes for address (vs 4 for IPv4)
  - 7 fields in header (vs 13 for IPv4)
  - better security (via authentication)
  - Type of service support

IPv6 Header



- Priority: specify data traffic or real-time traffic
- Flow label: identify a stream of packets between two end nodes
- Next header: next extension header

IPv6 Header (cont'd)

- What's different from IPv4 header
  - Larger address space: $7*10^{23}$ IPs / m^2
  - No fragmentation at the router. Only source can do it
  - No header checksum
- Extension headers
  - Support very large packet, called jumbogram
  - Source routing up to 24 hops
  - Fragmentation
  - Security
    - Authentication and Integrity: Use secret key and MD5 checksum
    - Encryption: Use DES-CBC algorithm