BEng Course B38CN: Introduction to Communications and Networks
Chapter 5. The Network Layer

(handwritten annotation, left)
Transport
Network ← CH5 packets
CH3 → LLC
CH4 → MAC } DL = frames
PHY ← bits...
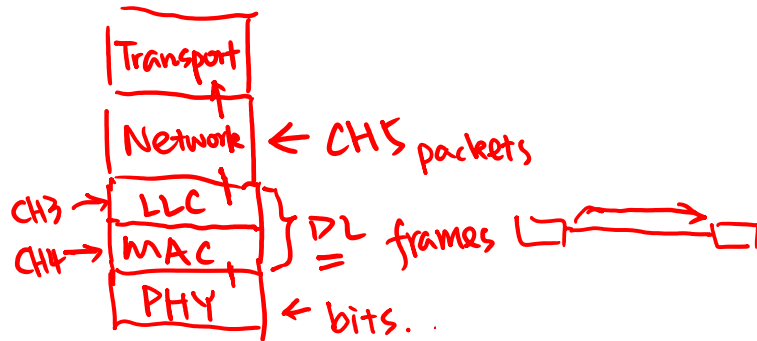
Sheng Tong

Xidian University

School of Telecommunications Engineering

Room: I-304, Main Building, North Campus
E-mail: ts_xd@163.com

Contents (1/2)

路由

Contents (2/2)

5.3 Congestion Control 拥塞

    5.3.1 General Principles of Congestion Control

    5.3.2 Congestion Prevention Policies

5.4 The Network Layer in the Internet

    5.4.1 The IP Protocol

    5.4.2 IP Addresses

5. The Network Layer

routers + links

- Provides services to the transport layer **through the communication subnet** by using the services provided by the data link layer.

- The **data link layer** knows only about the two stations at either end of the line.

- The **network layer** must know about the **topology of the subnet** and choose appropriate paths (routing) through it.

  路由

- **Routing**: determines how packets are routed (selecting paths across a network) from source to destination.

- **Congestion control**: controls traffic congestion between nodes.

- **QoS**: reliability, delay, jitter, and bandwidth.

  quality of service

- **Internetworking**: transfers packets from one network to another; deals with differences in addressing, packet size, and protocols etc.

  网络互联

5.1 Network Layer Design Issues

- The services provided to the transport layer and the internal design of the subnet.

5.1.1 Services Provided to the Transport Layer

- **Design goals** of the network layer services:
  - The services should be independent of the router technology.
  - The transport layer should be shielded from the differences of the routers.
  - The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

- Distinguish between **connection-oriented** and **connectionless** services:
  - **Virtual Circuit (VC)**: A route is chosen between sender and receiver. All packets travel the same route. Each packet carries an identifier (virtual circuit number) telling which virtual circuit it belongs to. Example: **ATM**.
  - **Datagram**: Packets are routed independently. No advance setup is needed (connectionless). Example: **Internet**.

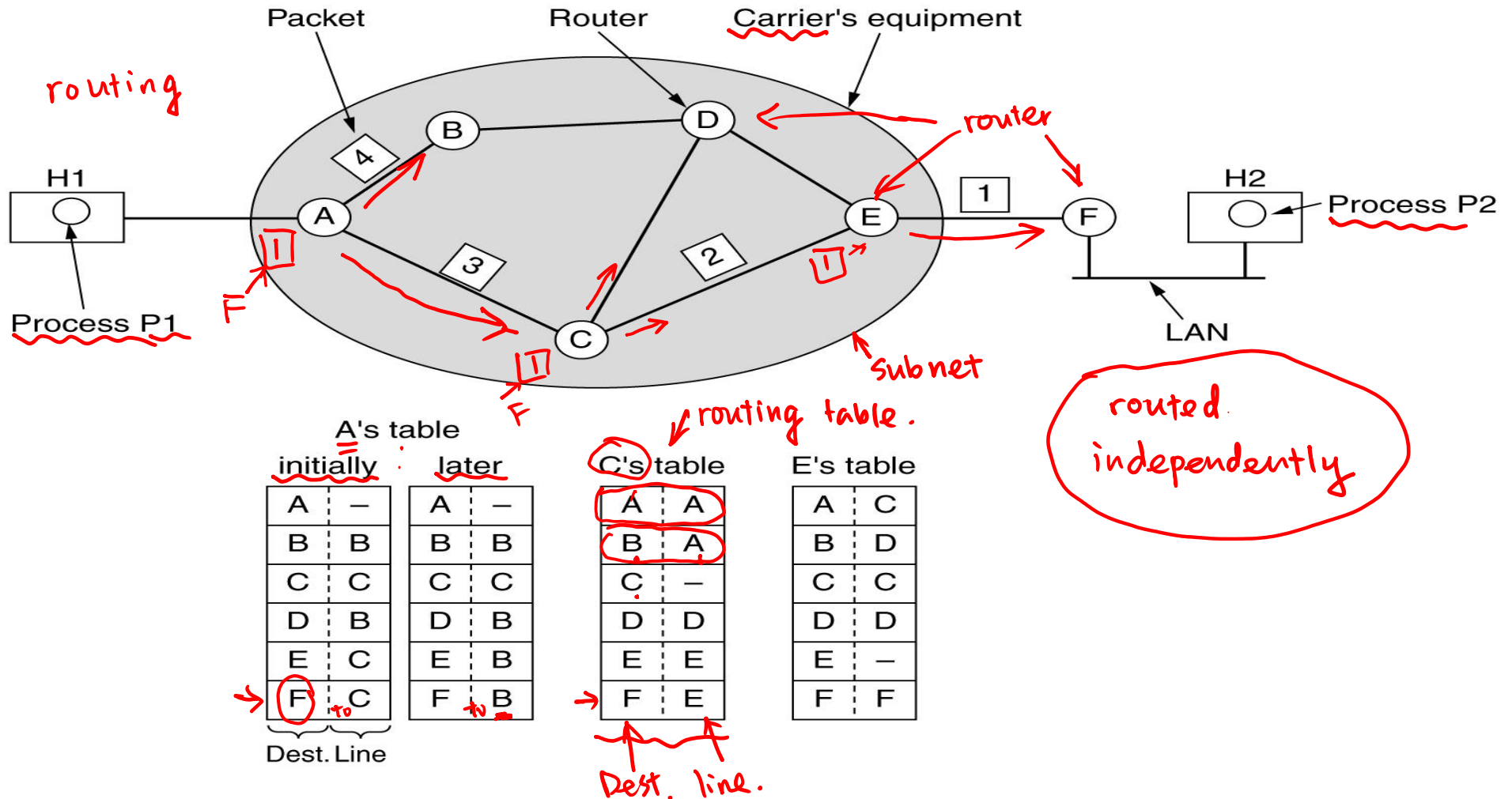# Implementation of Connectionless Service

Fig. 5.1: Routing within a datagram subnet.
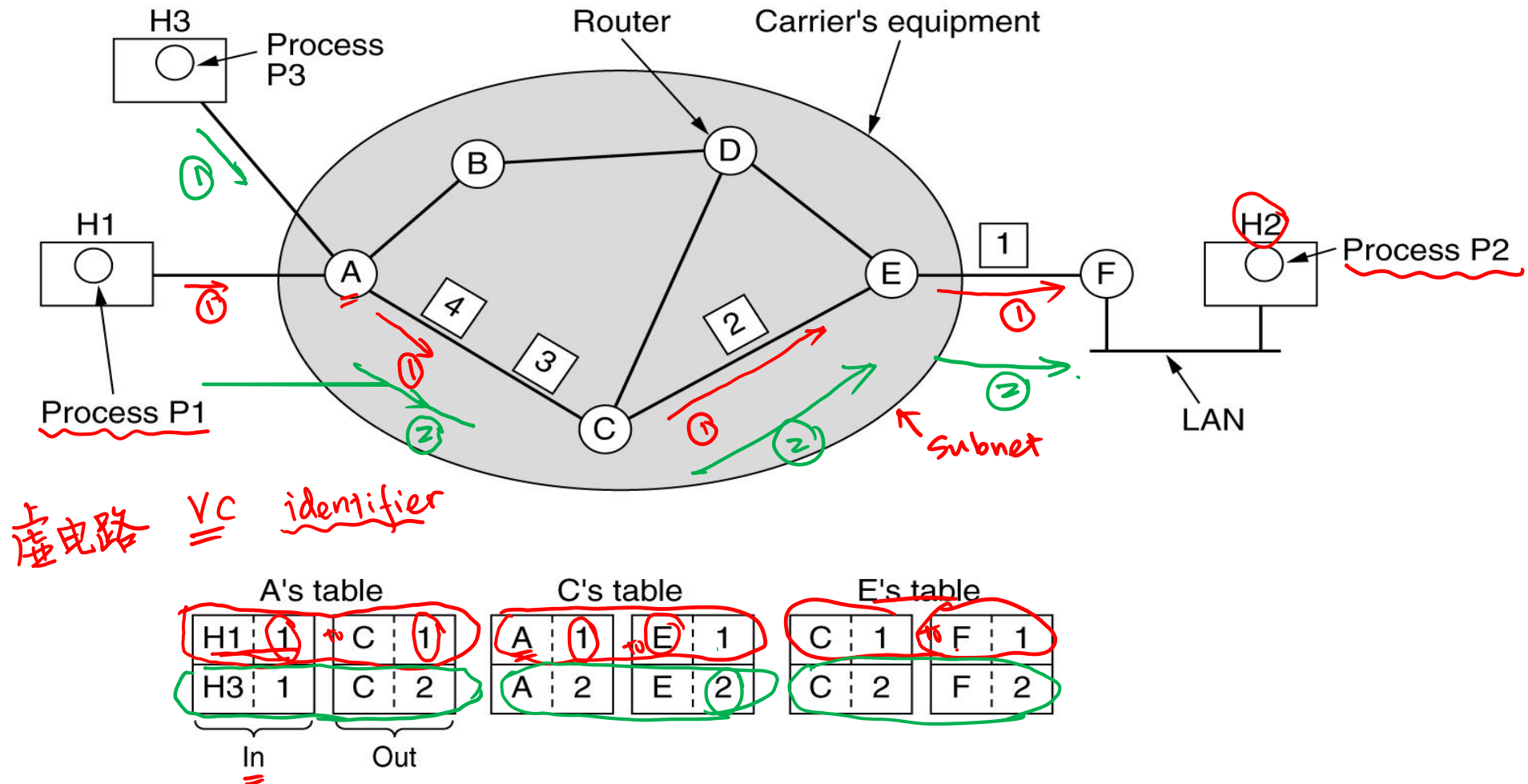
Implementation of Connection-Oriented Service

Fig. 5.2: Routing within a virtual-circuit subnet.

5.1.2 Comparison of Virtual-Circuit and Datagram Subnets

| Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address *(← independ routing)*	Each packet contains a short VC number
State information	Routers do not hold state information about connections *(no)*	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

5.2 Routing Algorithms

- Responsible for deciding which output line an incoming packet is to be sent.

- For **datagrams**, the route is deterrmined for **every** arriving data packet.

- For **virtual circuits**, the route is determined only **once**-at virtual circuits set up. Thereafter, data packets just follow the previously established route.

- **Two major classes**:

  - **Nonadaptive algorithms** (**static routing**): Routing decisions are made regardless of the current traffic and topology. The route is computed in advance. **Examples**: shortest path routing, flooding.

  - **Adaptive algorithms (dynamic routing)**: Routing decisions are made to reflect changes in the topology and the traffic. **Examples**: distance vector routing, link state routing.

5.2.1 The Optimality Principle

- **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 route.
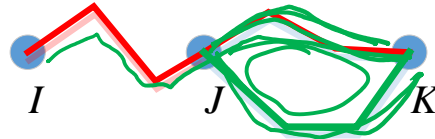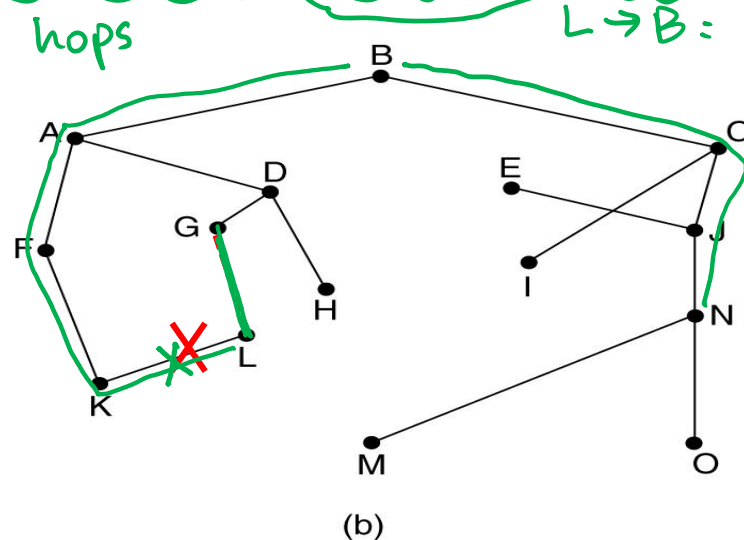
$$Red(I \to K) > G(J \to K) \; \times$$
$$Red(I \to J) + G(J \to K)$$
$$< Red(J \to J) + Red(J \to K)$$

- The set of optimal routes from all sources to a given destination form a tree rooted at the destination – **sink tree**.

  – Not necessarily unique.

  – The **goal** of all routing algorithms is to discover and use the sink trees for all routers.

hops

L → B : 4 hops
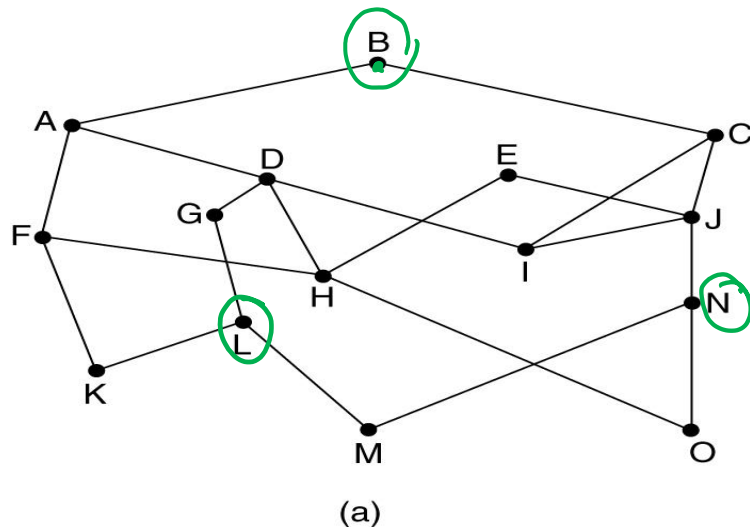
Fig. 5.3: (a) A subnet. (b) A sink tree for router B.

5.2.2 Shortest Path Routing

- Build a **graph of the subnet**, where each graph node is a router and each arc is a communication line/link.

- **Shortest path metric** (labels on the arcs): the number of hops, physical distance, or the mean queuing and transmission delay.

- **Labeling Algorithm**: *Dijkstra, 1959*

  - **Initially**, all nodes are labeled with **infinity** since no paths are known.

  - A label may be either tentative or permanent. Initially, all labels are **tentative**. When it is discovered that a label represents the **shortest possible path** from the source to that node, it is made **permanent** and never changed thereafter.

  - **Start** by marking the **source node** as permanent and consider it the **working node**.

  - Examine all the nodes **adjacent to** the working node. If the **sum** of the label on the working node and the distance from the working node to the considered node is **less than** the label on that node, we have a shorter path, so the node is **relabeled tentatively**.

  - Examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label **permanent**. This one becomes the new **working node**.
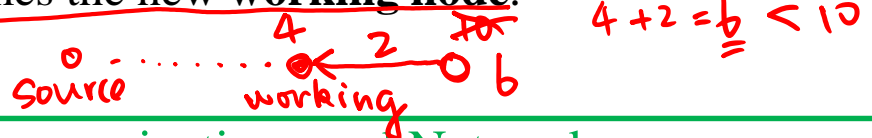
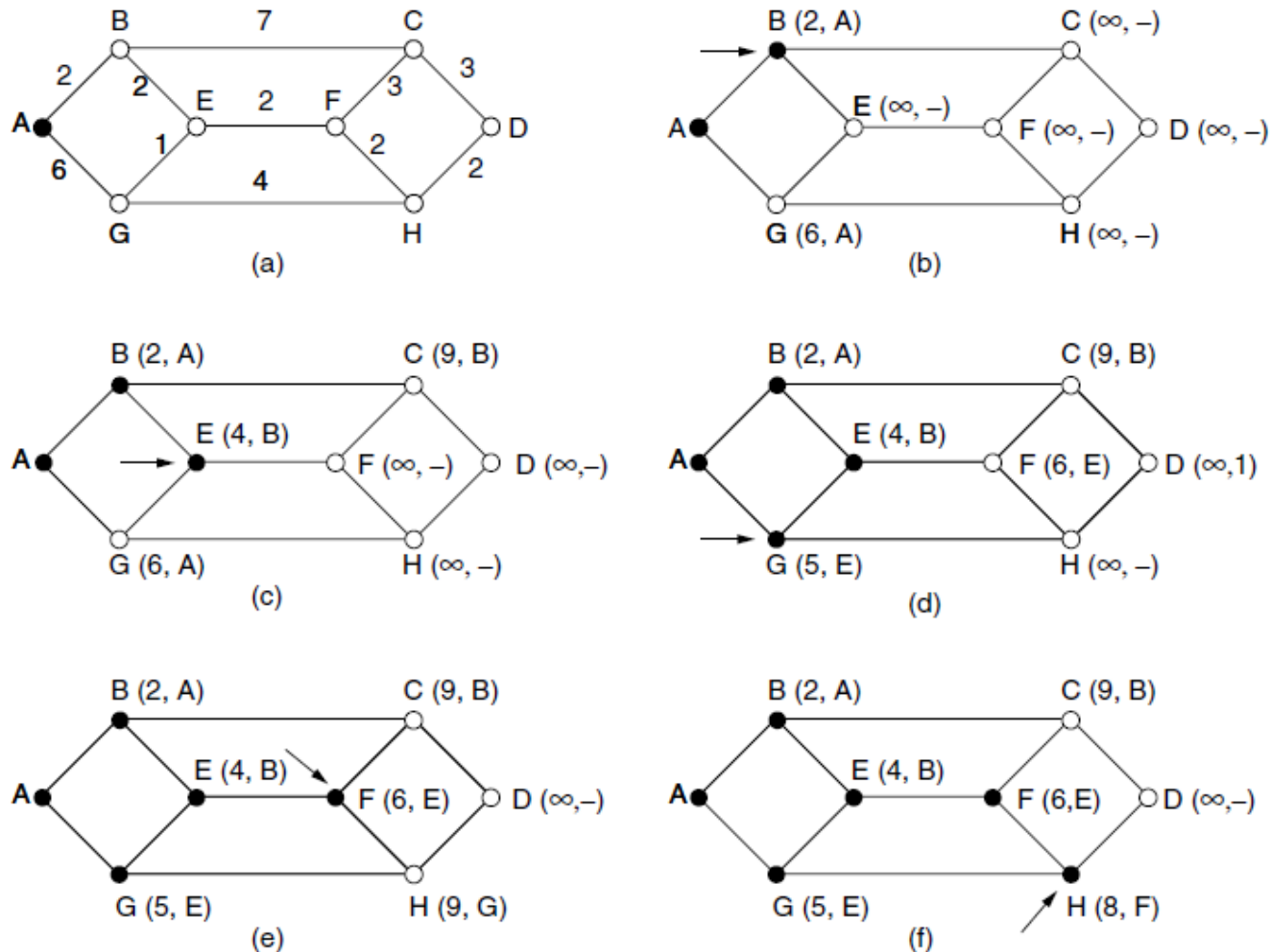# An Example of the Shortest Path Routing

Fig. 5.4: The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.

5.2.3 Flooding

- **Basic idea:** Forward an incoming packet across every outgoing line, except the one it arrived on.

  ⇨ Flooding always chooses the **shortest path**.

- **Problem:** Vast (infinite) numbers of duplicate packets are generated.
  - Use a **hop counter**: Decremented at each hop, with the packet being discarded when the counter reaches zero.
  - Keep track of which packets have been flooded, to avoid sending them out a second time. Require the source router to put a sequence number in each packet. Each router keeps track of the last sequence number per source router.
  - **Selective flooding**: Packets are sent in roughly the right direction.

- **Uses:** Flooding makes sense when **robustness** is needed, e.g., military applications or when all the data bases need to be updated concurrently.
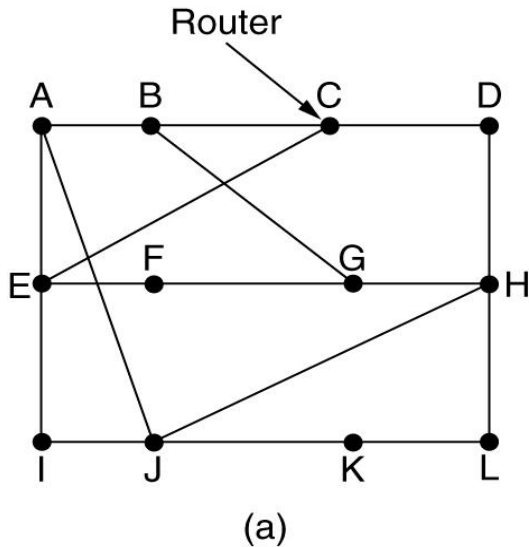
5.2.4 Distance Vector Routing

- Original ARPANET routing algorithm.

- Each router maintains a **routing table** (i.e., a vector) indexed by, and containing one entry for, each router in the subnet. This entry contains two parts:

  - The preferred outgoing line to use for that destination.

  - An estimate of the time or distance to that destination.

- **Basic idea**: Take a look at the costs that your direct neighbors are advertising to get a packet to the destination. Select the neighbor whose advertised cost, added with the cost to get to that neighbor, is the lowest. Advertise that new cost to the other neighbors.

An Example of Distance Vector Routing

Fig. 5.5: (a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

The Count-To-Infinity Problem

- Reacts quickly to **good news**, which travels one hop at a time.

- Reacts slowly to **bad news**, which takes longer because there is always a better route - until the whole network discovers the truth.

(a)

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

(b)

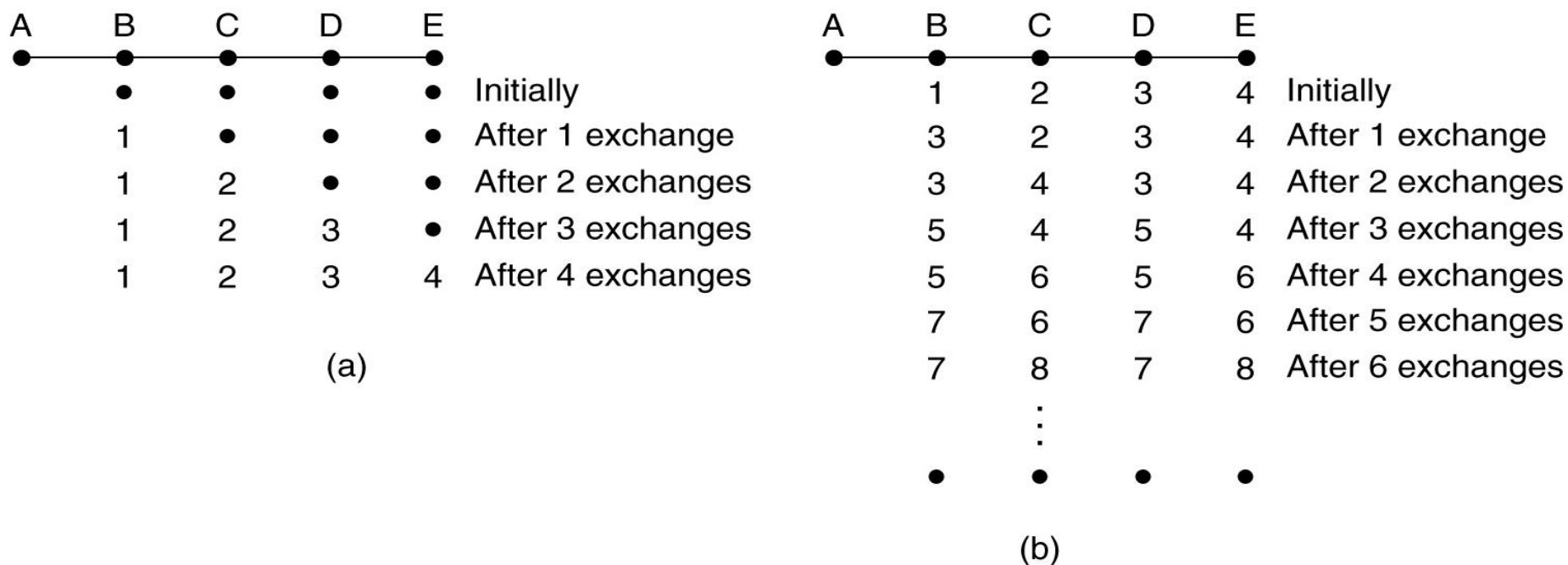| A	B	C	D	E	
•	•	•	•	•	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
	:	:	:	:	
	•	•	•	•	

Fig. 5.6: (a) Reacts quickly to good news. (b) Reacts slowly to bad news: the count-to-infinity problem.

5.3 Congestion Control

- When the number of packets dumped into the subnet is **within its carrying capacity**, they are all delivered. The number delivered is **proportional** to the number sent.

- As traffic increases too far, the routers are no longer able to cope and they **begin losing packets**.

- At very high traffic, **performance collapses** completely and almost no packets are delivered. This situation is called **congestion**.

- **Reasons for congestion**:
  – Insufficient link capacity - queues build up and insufficient memory.
  – Routers' CPUs (Nodes) are slow in processing packets.
  – Low-bandwidth lines.
  ⇨ The real problem is often a **mismatch** between parts of the system.
  ⇨ Upgrading part, but not all, of the system, often just moves the bottleneck somewhere else.
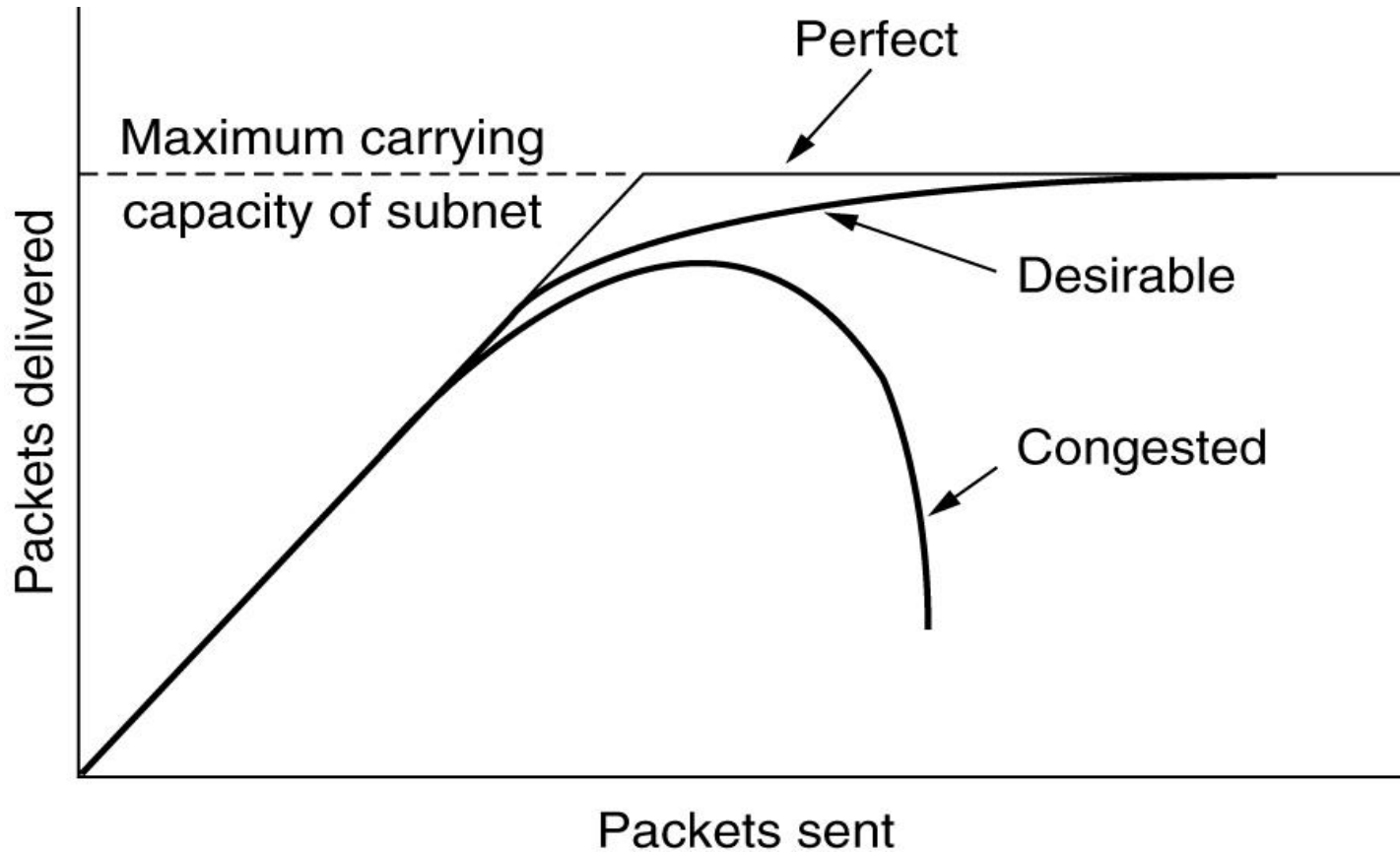
Congestion Symptom

Fig. 5.7: When too much traffic is offered, congestion sets in and performance degrades sharply.

Congestion Control vs. Flow Control

- **Flow control** relates to the **point-to-point** traffic between a given sender and a given receiver. Its job is to make sure that a fast sender will not swamp a slow receiver.

- **Congestion control** is to make sure the **subnet** can carry the offered traffic. It is a **global issue**.

- **Example 1**: Consider a fiber optic network with a capacity of 1000 Gbps on which a supercomputer is trying to transfer a file to a personal computer at 1 Gbps. Although there is no congestion control (network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer a chance to breathe.

- **Example 2**: Consider a store-and-forward network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the other half. Here, since the total offered traffic exceeds what the network can handle, congestion control is necessary. No flow control is needed since the problem is not that of fast senders overpowering slow receivers.

5.3.1 General Principles of Congestion Control

- **Open loop control**: makes decisions without regard to the current state of the network. Attempts to solve the problem by good design in essence and avoid the congestion in the first place. Once the system is up and running, midcourse corrections are not made.

- **Closed loop control**: requires feedback. Three parts:
  - Monitor the system to detect when and where congestion occurs.
  - Pass information to places where action can be taken.
  - Adjust system operation to correct the problem.

- The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle. ⇨**Two solutions**:
  - **Increase the resources**, e.g., increase the bandwidth or use spare routers.
  - **Decrease the load**, e.g., deny or degrade service to some even all users.

5.3.2 Congestion Prevention Policies

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Fig. 5.8: Policies that affect congestion.

5.4 The Network Layer in the Internet

- At the network layer, the Internet can be viewed as **a collection of subnetworks** or **Autonomous Systems (ASes)** that are interconnected.

- The **glue** that holds the whole Internet together is the network layer protocol, **IP (Internet Protocol)**.

- **Communication in Internet** works as follows:

  - The transport layer takes data streams and breaks them up into datagrams.

  - Each datagram is transmitted through the Internet, possibly being fragmented into smaller units as it goes.

  - When all the pieces finally get to the destination machine, they are reassembled by the network layer into the original datagram.

  - This datagram is then handed to the transport layer.
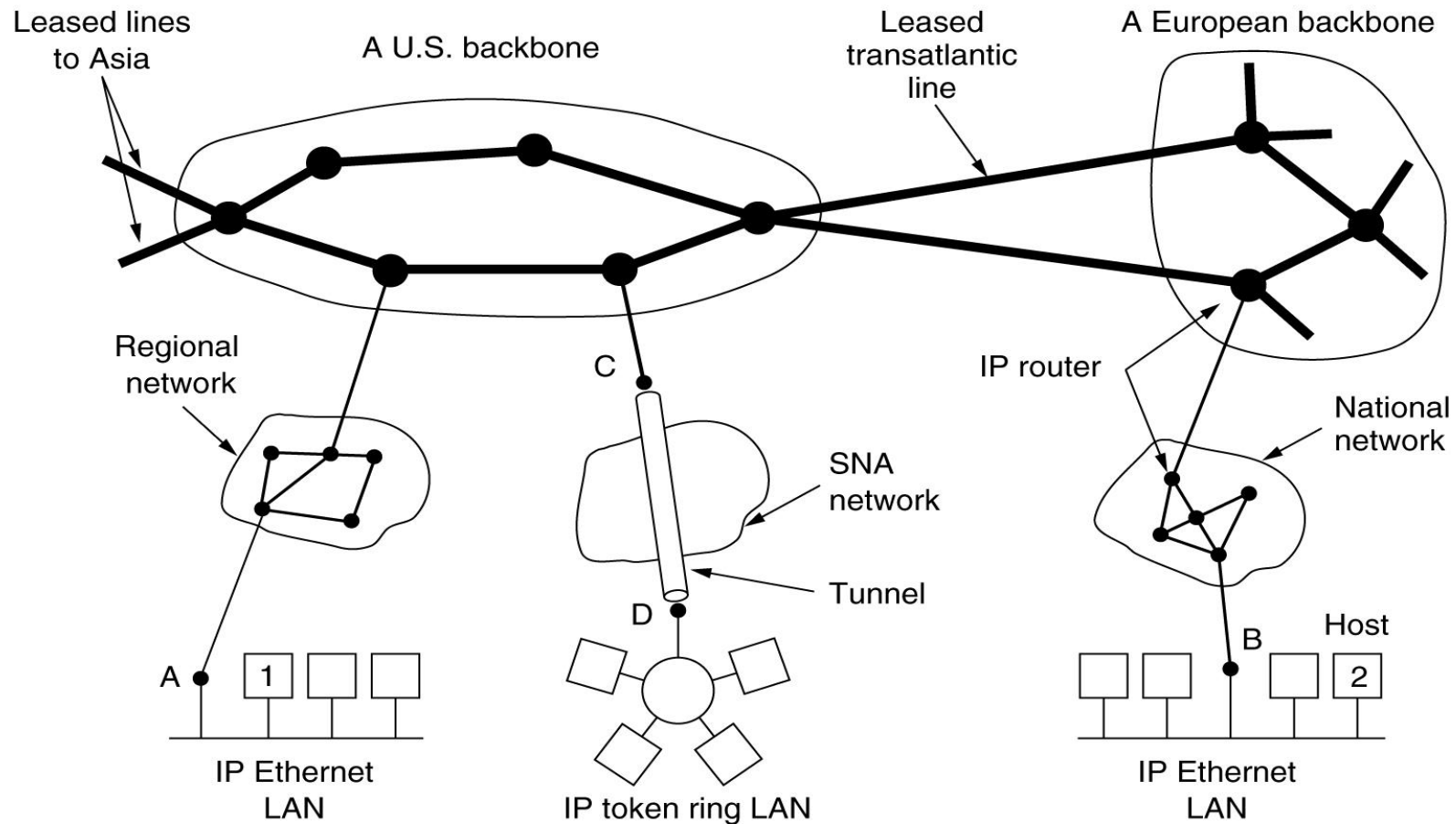
Collection of Subnetworks

Fig. 5.9: The Internet is an interconnected collection of many networks.

5.4.1 The IP Protocol

• An **IP datagram** consists of a header and data (from transport layer).

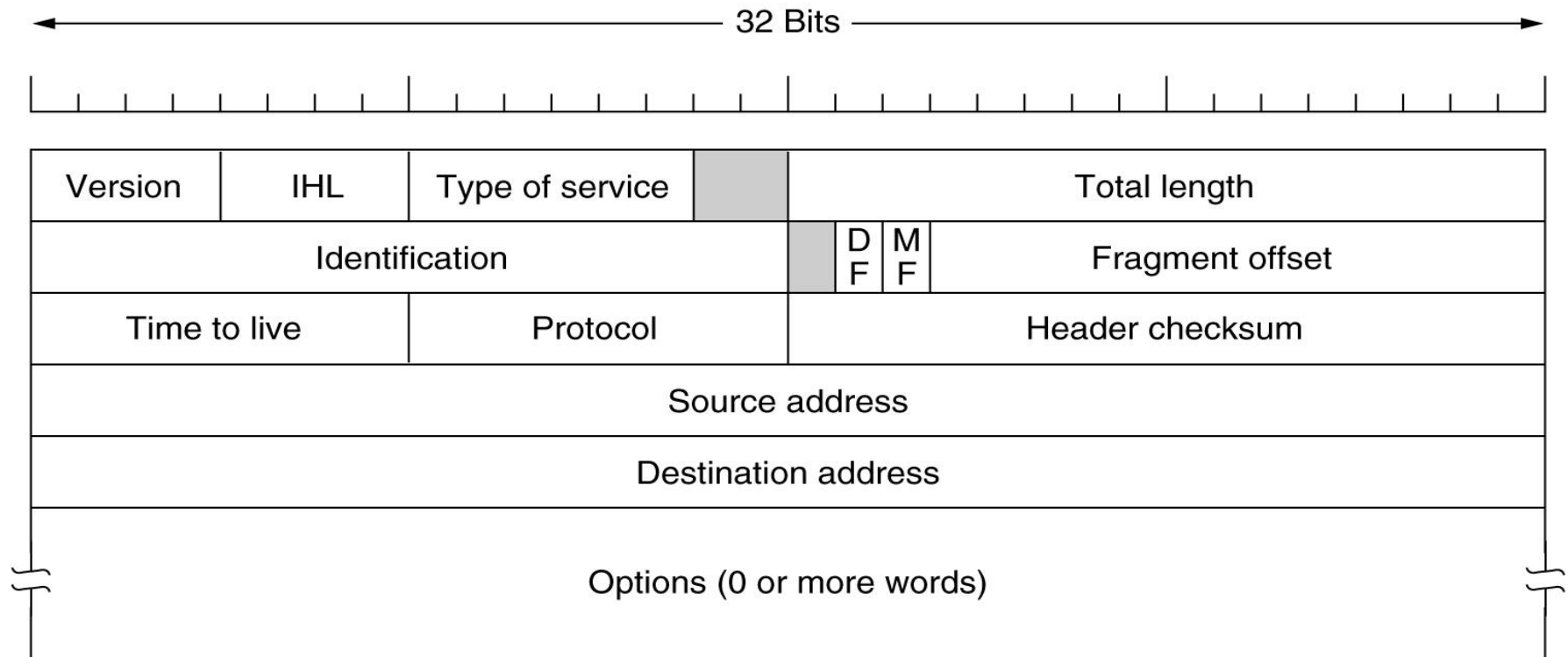– The header has a **20-byte fixed** part and a **variable** length **optional** part.

Fig. 5.10: The IPv4 (Internet Protocol) header.

IP Datagram Header

- **Version**: 4-bit; identifies the version of IP (IPv4 or IPv6) used by the datagram.

- **IHL**: 4-bit; **IP H**eader **L**ength in 32-bit words; 5 (20 bytes)≤IHL≤15 (60 bytes).

- **Type of service**: 6-bit; specifies the choices of delay, throughput, and reliability.

- **Total length**: 16-bit; length of (header + data).

- **Identification**: 16-bit; determines which datagram a fragment belongs to.

- **DF**: 1-bit; **D**on't **F**ragment.

- **MF**: 1-bit; **M**ore **F**ragments.

- **Fragment offset**: 13-bit; tells where in this datagram the fragment belongs to.

- **Time to live**: 8-bit; used to limit packet lifetimes in the network.

- **Protocol**: 8-bit; transport protocol (TCP or UDP) used.

- **Header checksum**: 16-bit; for header only.

- **Source/Destination addresses**: 32-bit; full network addresses.

IP Datagram Header Options

- The **options field** was designed to provide an escape to allow subsequent versions of the protocol to include information not present in the original design, to permit experimenters to try out new ideas, and to avoid allocating header bits to information that is rarely needed.
  - **Variable** length.
  - Padded out to a **multiple of four 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	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Fig. 5.11: Some of the IP options.

5.4.2 IP Addresses

- Every host and router on the Internet has an IP address.

- It encodes its network number and host number.

- All IP addresses are **32 bits** long and are used in the Source address and Destination address fields of IP packets.

- **Classful addressing**: IP addresses were divided into **five categories**.
  - **Class A**: 7-bit Network No., 24-bit Host No. Up to **128** networks with 16 million (16,777,216) hosts each.
  - **Class B**: 14-bit Network No., 16-bit Host No. Up to 16384 networks with 64K (65536) hosts each.
  - **Class C**: 21-bit Network No., 8-bit Host No. Up to 2 million (2,097,152) networks with **256** hosts each.
  - **Class D**: 28-bit Multicast address; Datagram broadcasts to multiple hosts.
  - **Class E**: 28-bit; reserved for future expansion.

Five Categories of IP Addresses

- An IP address is written in decimal as four numbers (8 bits), separated by dots, which is known as dotted decimal notation.

- The lowest IP address is 0.0.0.0 and the highest is 255.255.255.255.
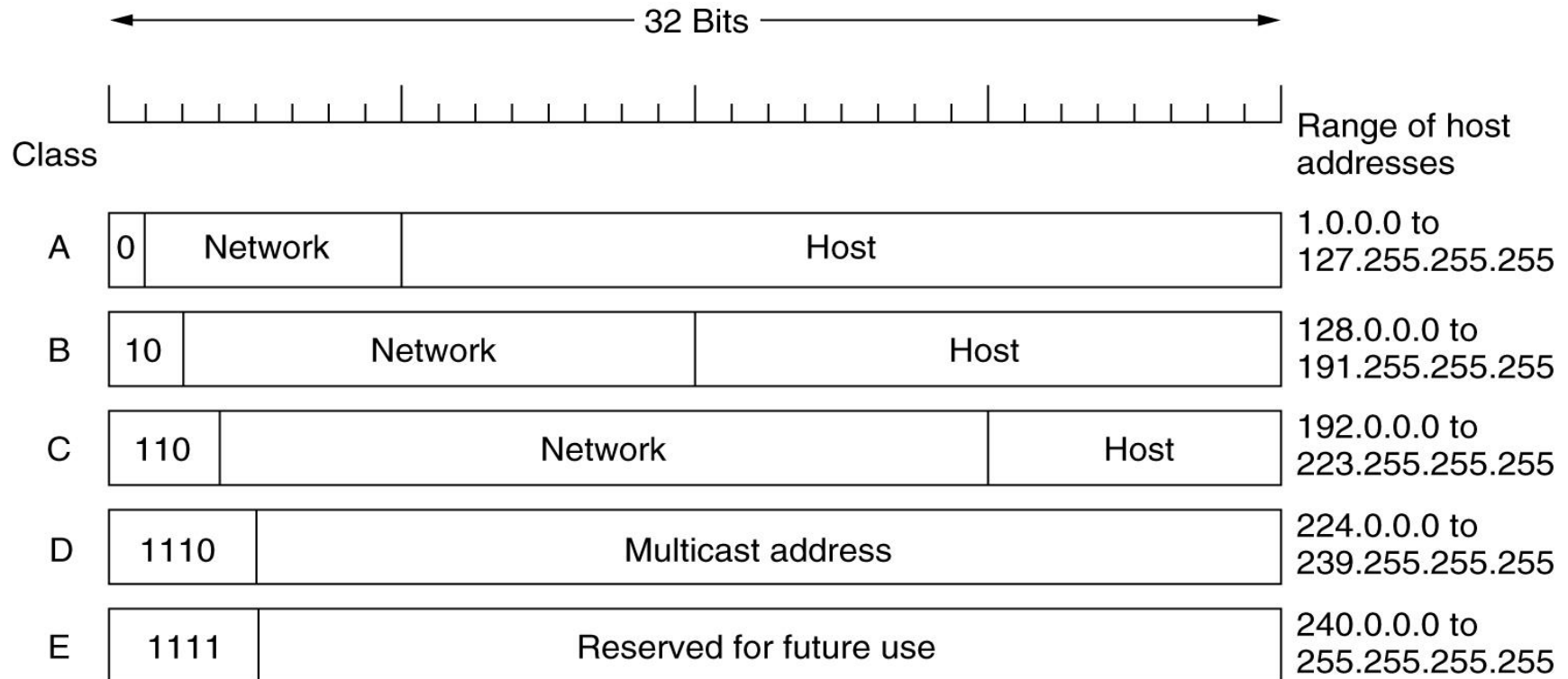

Fig. 5.12: IP address formats.

Special IP Addresses

- **0.0.0.0**: used by hosts when they are being booted.

- **All 0s as network number**: the current network.

- **255.255.255.255**: broadcast to all hosts on the indicated network (typically a LAN).

- **All 1s as host number**: broadcast packets to distant LANs anywhere in the Internet.

- **127.xx.yy.zz**: loopback testing.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	This host
0 0 . . . 0 0 │ Host	A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Broadcast on the local network
Network │ 1 1 1 1 . . . 1 1 1 1	Broadcast on a distant network
127 │ (Anything)	Loopback

Fig. 5.13: Special IP addresses.

Subnets

- **Problem:** All the hosts in a network must have the same network number. This is due to the **rule** that a single class A, B, or C address refers to one network, not to a collection of LANs. This property may cause a single organization to acquire several classes of addresses.

- **Solution**: Allow a network to be split into several parts ("**subnets**") for internal use but still act like a single network to the outside world.

  - "**Subnet**" here (exclusively) means a local partition, not the traditional concept as the set of all routers and communication lines in a network.

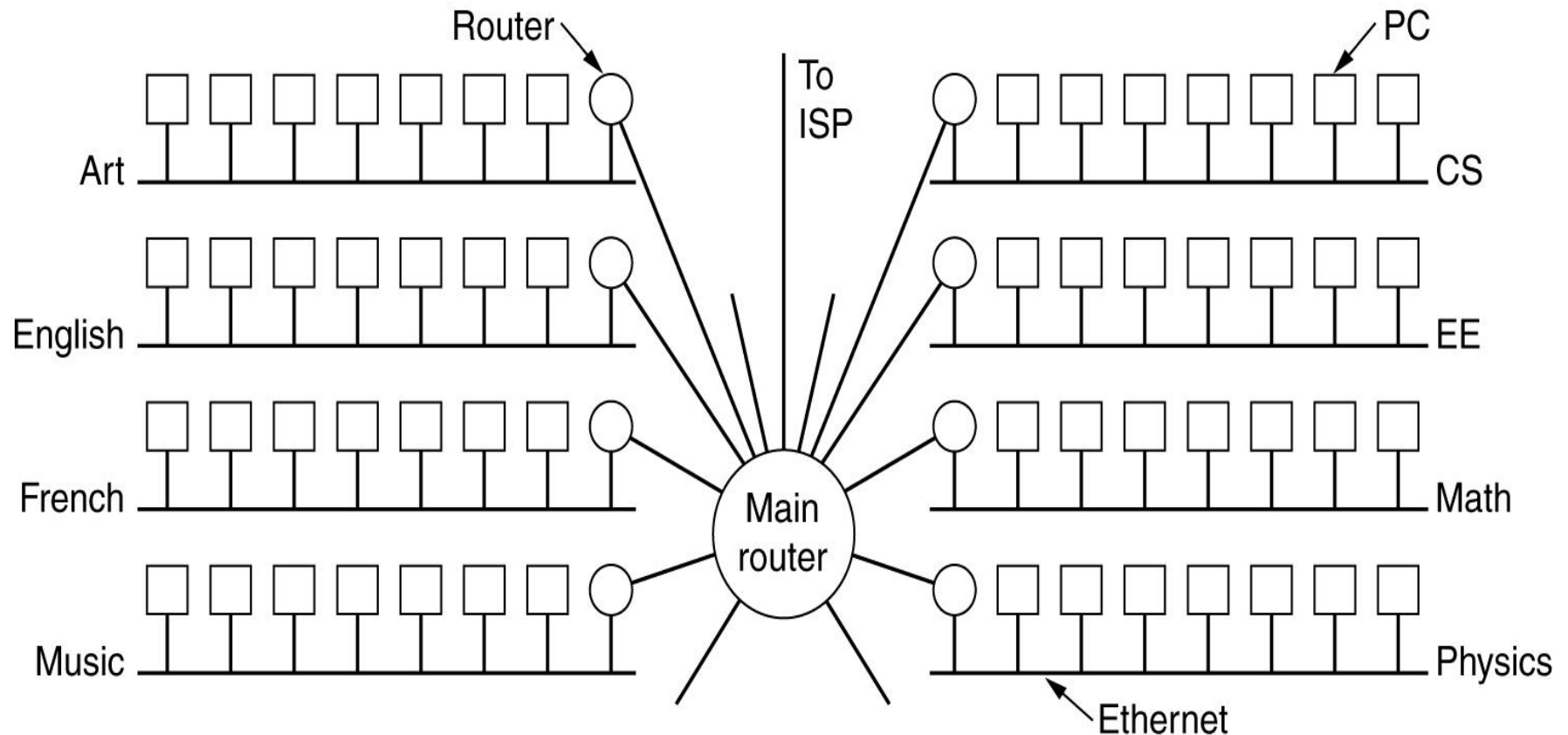# A Typical Campus Network with "Subnets"

Fig. 5.14: A campus network consisting of LANs for various departments.

Subnet Mask

- To **implement subnetting**, we use a single network address for the entire organization, and internally **divide** the host address space into a **subnet address** and a host address.

- **Subnet mask**: network + subnet number; also written in **dotted decimal** notation, with the addition of a slash followed by the number of bits in the (network + subnet) part.
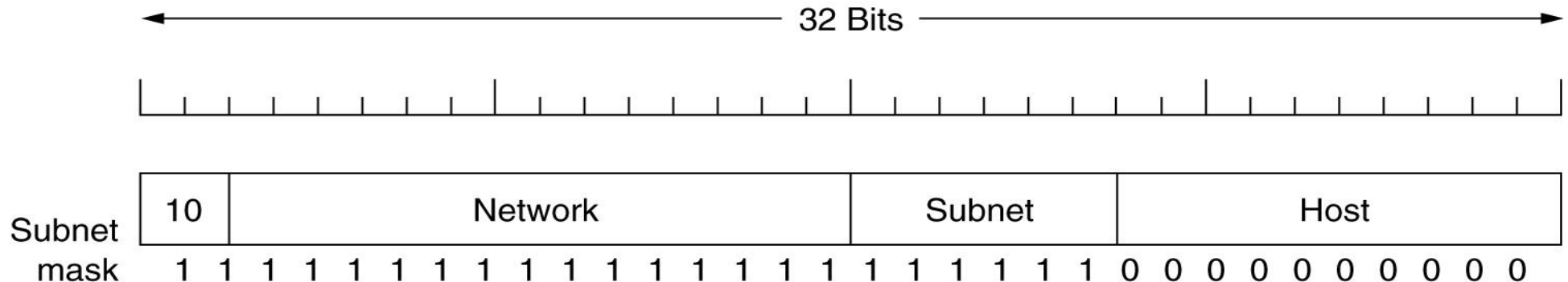
  - In the following figure, subnet mask: 1) 255.255.252.0; 2) /22.

Fig. 5.15: A class B network subnetted into 64 subnets.