Computer Networks

Dr. Xiqun Lu
College of Computer Science & Technology,
Hangzhou, China
Jan.2, 2024

INTRODUCTION

Network Hardware

- There is no generally accepted taxonomy into which all computer networks fit, but two dimensions stand out as important: **transmission technology** and **scale**.
- Broadly, *two types of transmission technology* are in widespread use
  - **Broadcast links** (Broadcasting广播, Multicasting多播)
    - A wireless network is a common example of a broadcast link
  - **Point-to-point links** (Unicasting单播)

Network Hardware

- An alternative criterion for classifying networks is **scale**.
- **Distance** is important as a classification metric because different technologies are used at different scales.

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Figure 1-6. Classification of interconnected processors by scale.

Network Hardware: Communication Links

- There are many types of communication links, which are made up of different types of physical media, such as
  - Twisted Pair (telephone)
  - Coaxial cable (TV)
  - Fiber optics (the backbone the PSTN, Public Switched Telephone Network)
  - Radio spectrum (cellphone)

- Different links can transmit data at different rates, with **the transmission rate** of a link measured in *bits/second*.

Network Software

- Network software is highly structured. The approach described here forms the keystone of the entire book and will occur repeatedly later on.
- **Protocol Hierarchies**
  - Most networks are organized as a stack of layers or levels, each one built upon the one below it.
  - The purpose of each layer is to offer certain **services** to the higher layers while *shielding* those layers from the details of how the offered services are actually implemented.
  - When layer *n* on one machine carries on a conversation with layer *n* on another machine, the rules and convections used in this conversation are collectively known as the layer *n* **protocol**.
    - A protocol is an agreement between the communicating parties on how communication is to proceed.

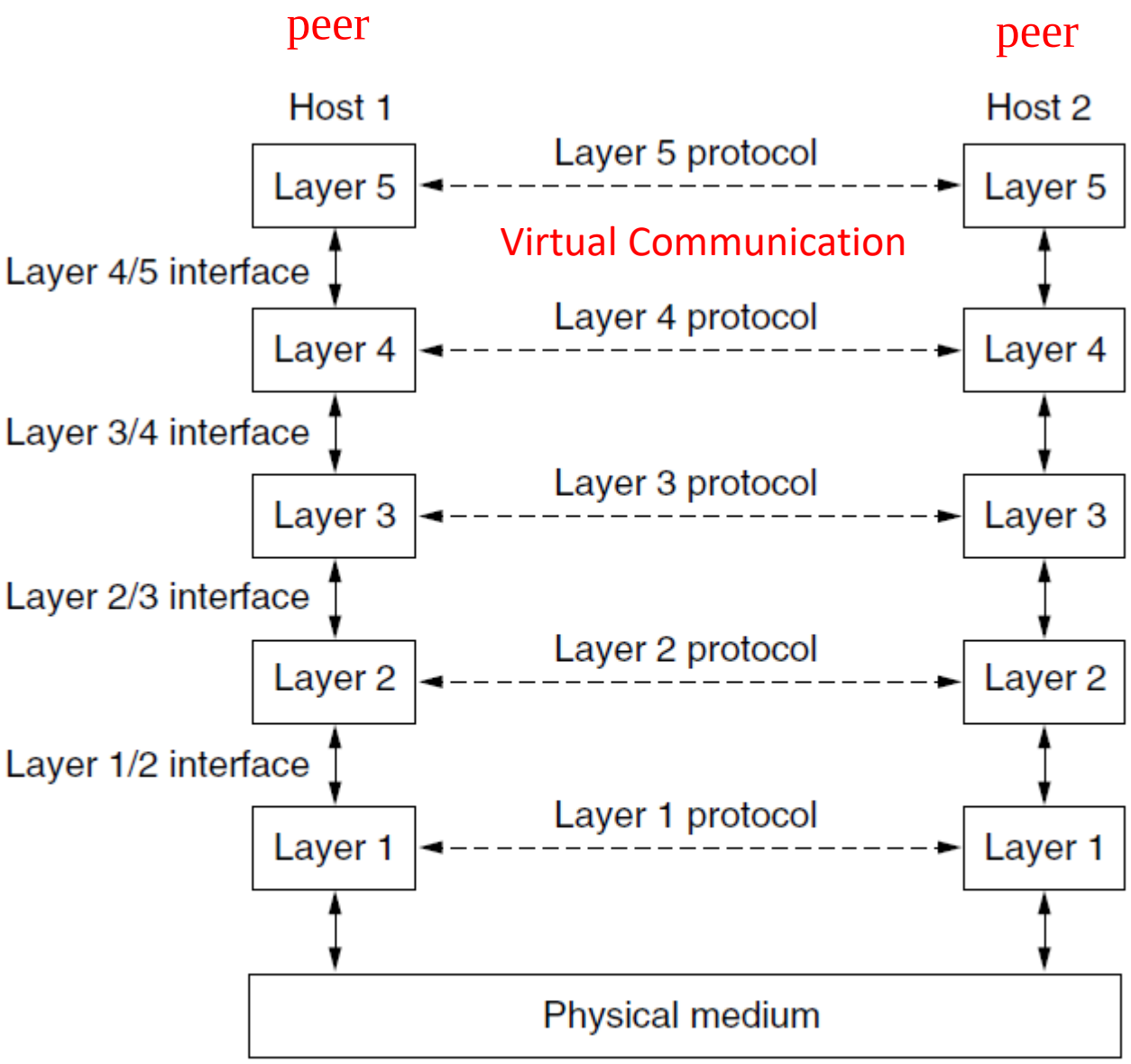


Figure 1-13. Layers, protocols, and interfaces.

1) Each layer passes data and control information to the layer immediately below it, until the lowest layer is reached.
2) The physical medium is where the actual communication occurs.
3) Between each pair of adjacent layers is an **interface**. The interface defines which **primitive operations** and **services** the lower layer makes available to the upper one.

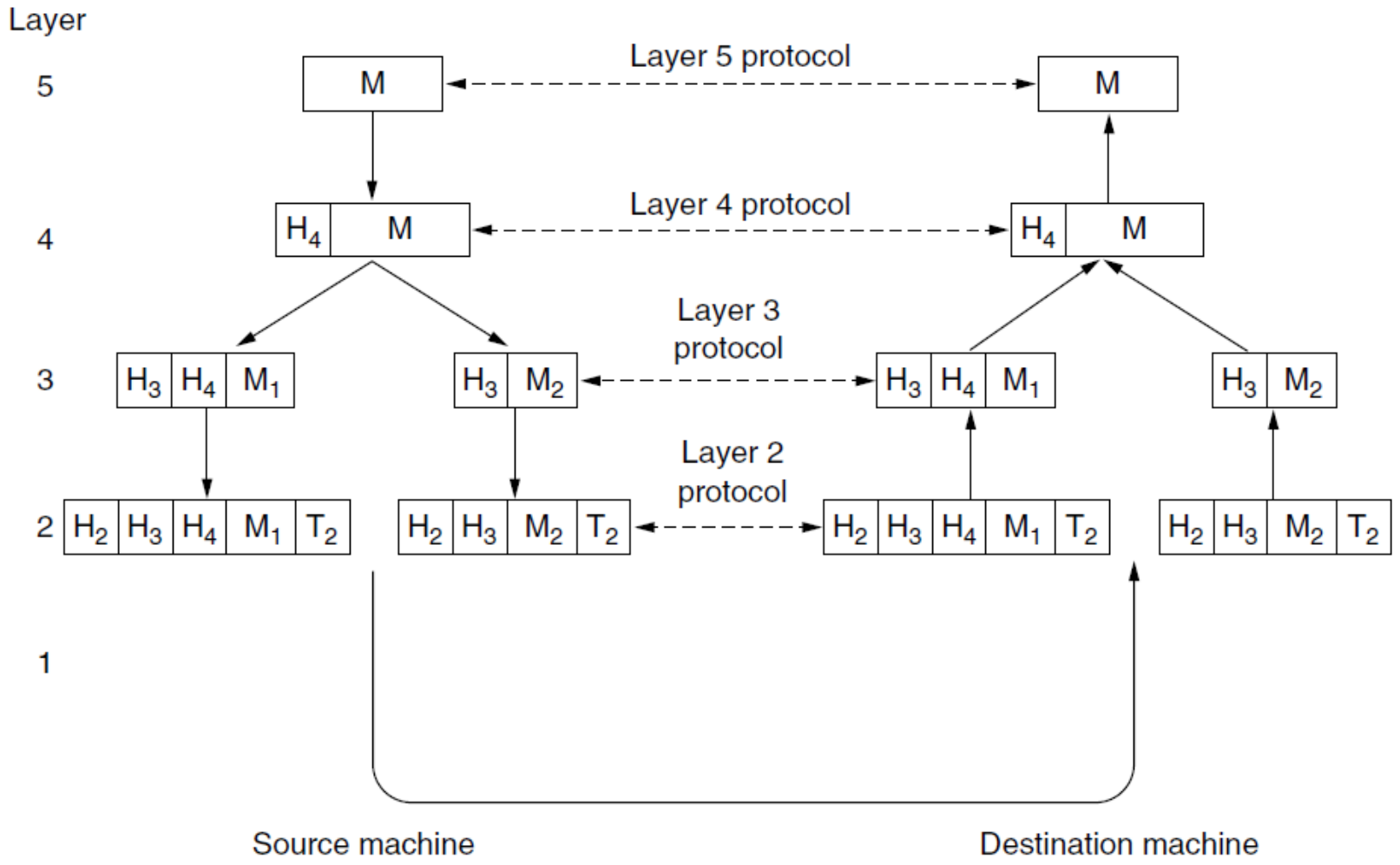


Figure 1-15. Example information flow supporting virtual communication in layer 5.

In many networks, layer 3 usually break up the incoming messages into smaller units, packets, prepending a layer 3 header to each packet.
Layer 2 adds to each piece not only a header but also a trailer.
The lower layers of a protocol hierarchy are frequently implemented in hardware or firmware.

OSI and the TCP/IP Reference Models

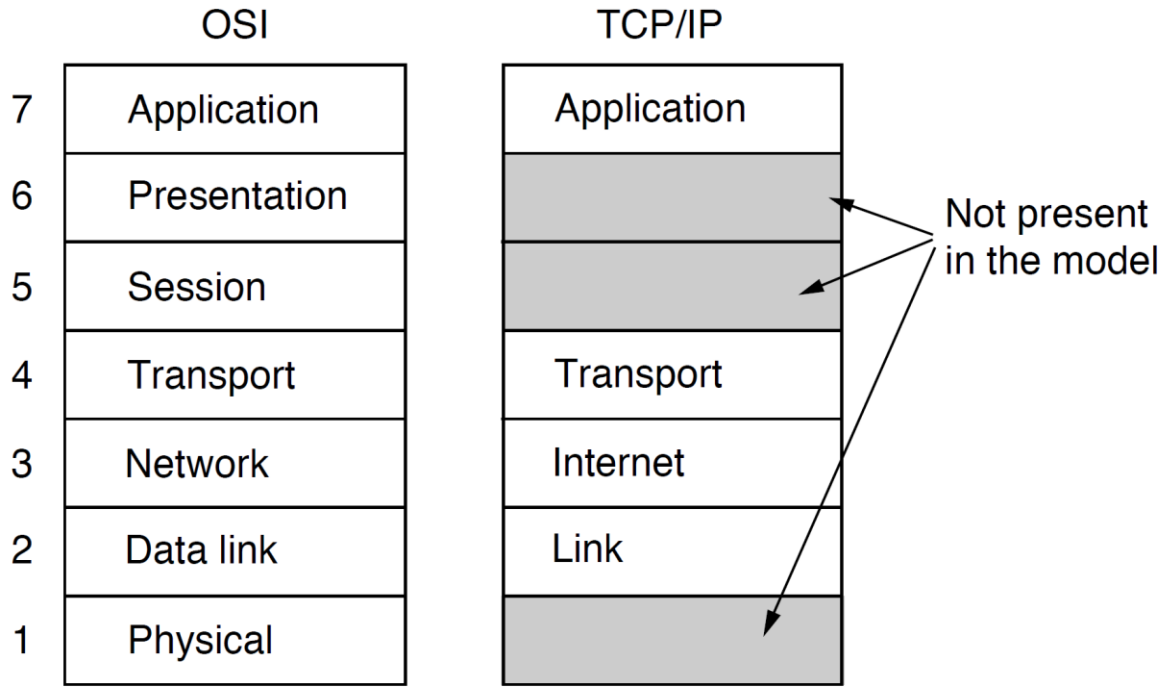


Figure 1-33. The TCP/IP reference model.

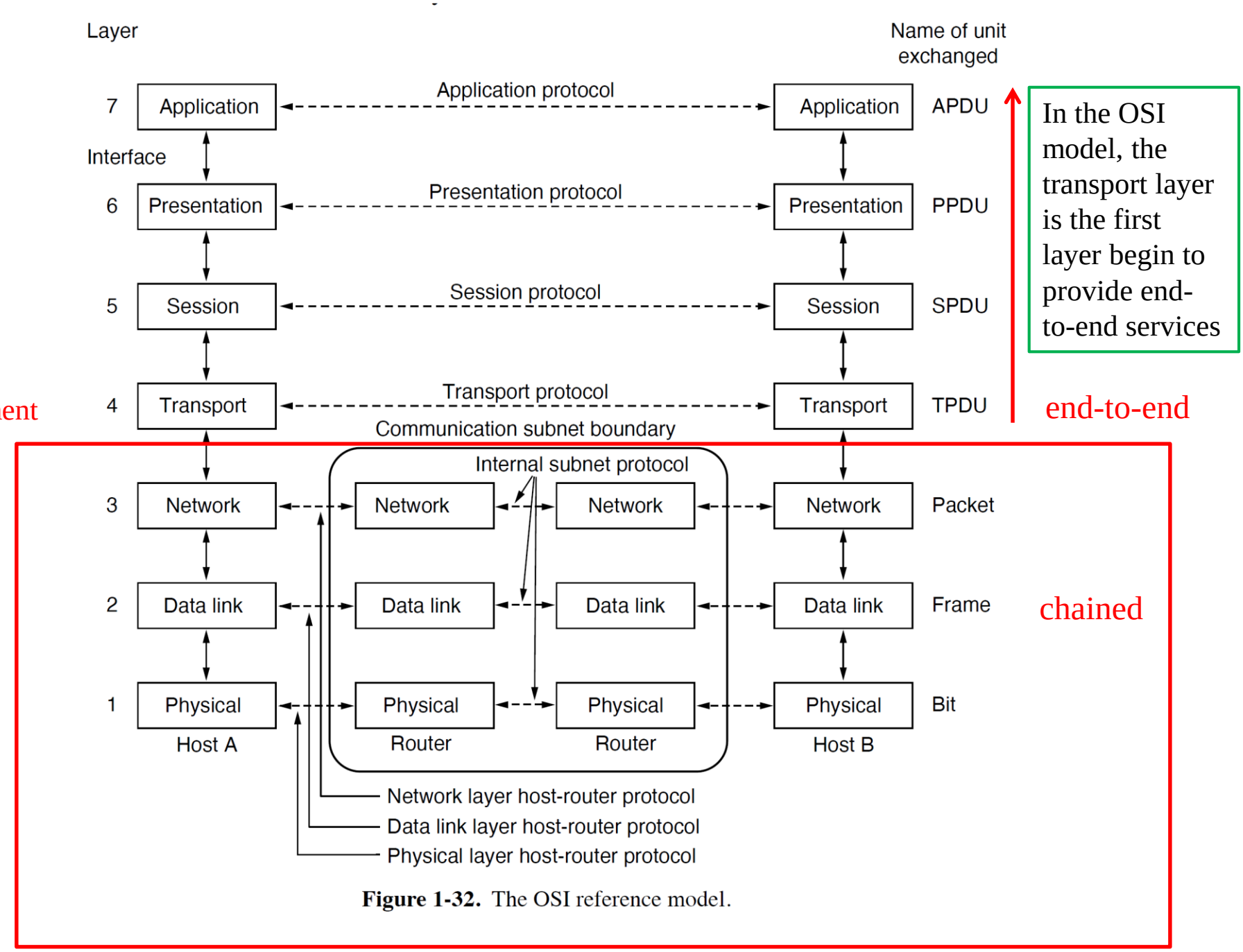


Figure 1-32. The OSI reference model.

The TCP/IP Reference Model

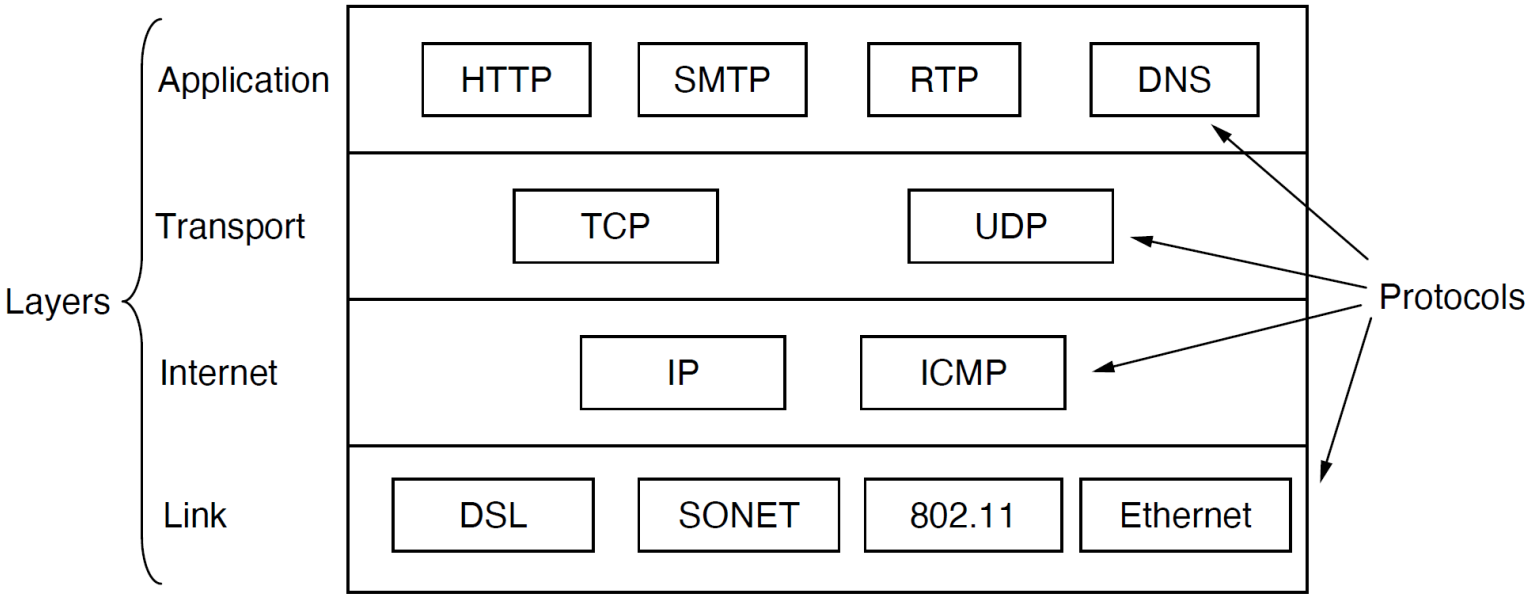


Figure 1-34. The TCP/IP model with some protocols we will study.

Introduction: Metric Units

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Figure 1-38. The principal metric prefixes.

Quantitative Metrics of the Performance of Packet-Switch Networks

- **Delay**
  - *Processing delay*: the time required to examine the packet's header and determine where to direct the packet
  - *Queuing delay*: depends on the number of earlier-arriving packets
  - *Transmission delay* (or *the store-and-forward delay*): L(the length of packet in bits)/R(the transmission rate of a link)
  - *Propagation delay* is the distance between two routers divided by the propagation speed.
- **Loss**: in reality a queue preceding a link has finite capacity.
- **Throughput**: depends on the transmission rate of *the bottleneck link* in the network.

PHYSICAL LAYER

Physical Layer: Data Rate vs. Bandwith

- Nyquist Bandwidth
  - Binary: C = 2B
  - Multilevel Signaling: **$C = 2B \log_2 V$** (V is the number of discrete signals or voltage levels)

- Shannon Capacity
  - $C = B \log_2(1+SNR)$

$$SNR_{dB} = 10\log_{10}(\text{signal power / noise power}) = 10\log_{10}(SNR)$$

- 如果两个公式都可以套用，那么Data Rate取计算出来小的那个值！

Physical Layer: Transmission Media

- Guided
  - Twisted Pairs
    - A signal is usually carried as **the difference in voltage** between the two wires in the pair. — immune to external noise
    - The wires **are twisted together in a helical form**. — cancelling out electromagnetic interference
  - Coaxial Cable (同轴电缆)
  - Power Lines
  - Optical Fibers
- Wireless
  - Multipath fading
- Satellite (转发)
  - GEO (同步卫星)

Physical Layer: Digital Modulation for Baseband Signals

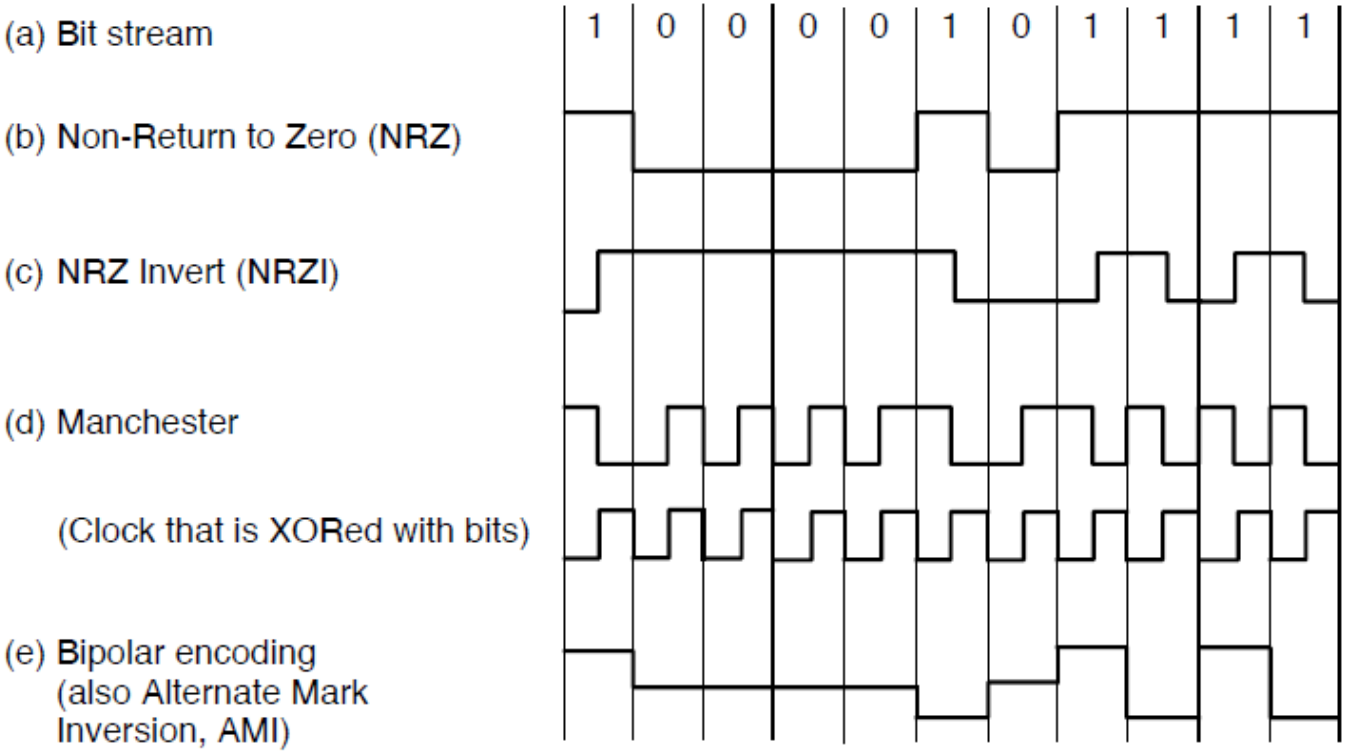


Figure 2-14. Line codes: (a) Bits, (b) NRZ, (c) NRZI, (d) Manchester, (e) Bipolar or AMI.

注意具体应用所采用具体的调制技术，如USB用的是NRZI，Ethernet采用的Manchester Encoding技术。

Physical Layer: Digital Modulation for Passband Signals

Binary ASK

Binary FSK

Binary PSK

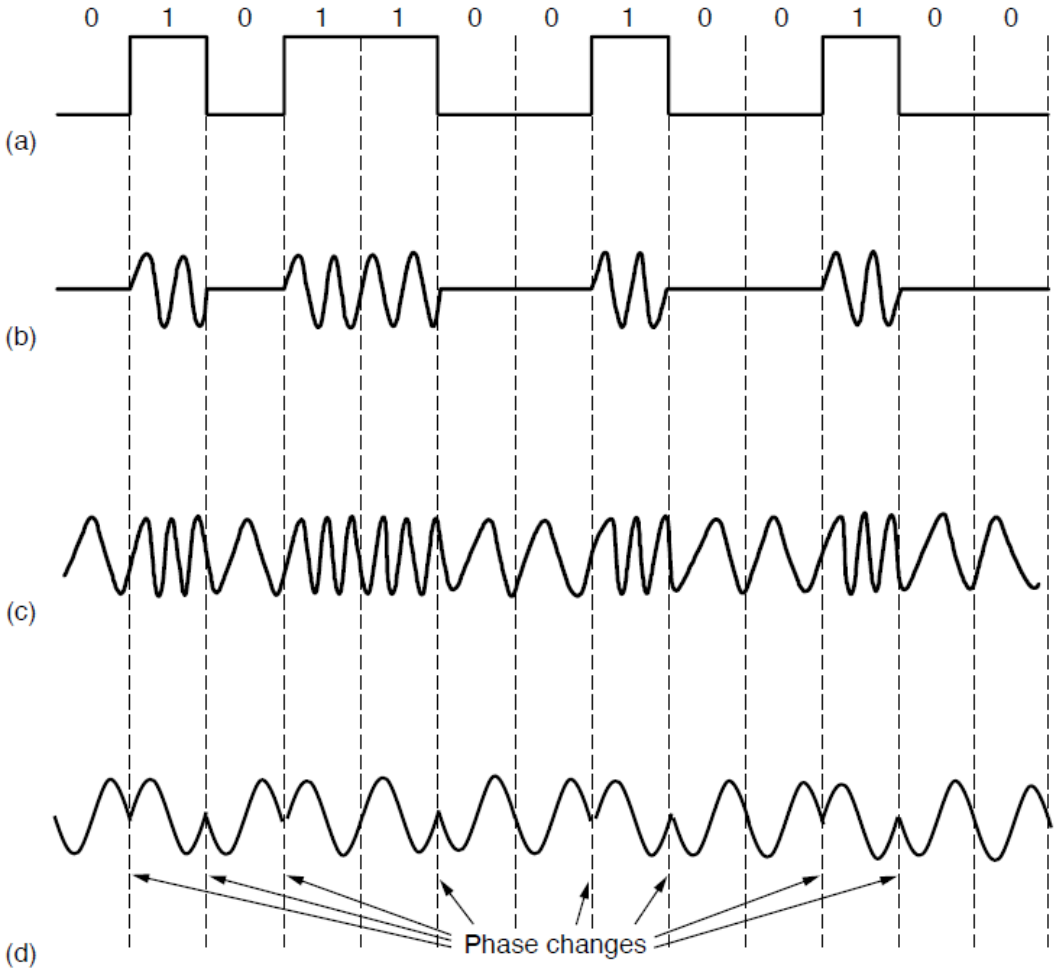


Figure 2-16. (a) A binary signal. (b) Amplitude shift keying. (c) Frequency shift keying. (d) Phase shift keying.

除了这些技术之外，还有更复杂QPSK，QAM-16，QAM-64。

Physical Layer: Multiplexing

- **FDM** (Frequency Division Multiplexing) (Fig. 2-25)
  - To divide the spectrum into frequency bands, with each user having exclusive possession of some band in which to send their signal.
  - OFDM (Fig. 2-26)
  - WDM (for optical fiber, $c = \lambda f$)
- **TDM** (Time Division Multiplexing) (Fig.2-27)
  - The users take turns (in a round-robin fashion), each one *periodically* getting the entire bandwidth for a little burst of time (time slot).
- **CDM** (Code Division Multiplexing) (Fig.2-28)
  - Each user's message is spread out over a unique chip sequence which allows each station to transmit over the entire frequency spectrum all the time.
  - In CDMA, each bit time is subdivided into *m* short intervals called **chips**.

Two Widely Deployed Communication Systems

- PSTN (Public Switch Telephone Network)
  - Circuit-switched
- Cellular network
  - Frequency reuse

DATA LINK LAYER

Data Link Layer

- Three main functions of the data link layer
  - Framing
  - Dealing with errors (error detection and error correction)
  - Flow control

Data Link Layer: Framing

- Framing (to find the start of new frames while using little of the channel bandwidth)
  - 1) Byte Count (Fig.3-3)
  - 2) Flag bytes with byte stuffing (Fig.3-4) (ESC, Flag)
    - PPP protocol
  - 3) Flag bits with bit stuffing (Fig.3-5) (five consecutive incoming 1 bits followed by a 0 bit)
    - Application example: USB
    - HDLC protocol
  - 4) Physical layer coding violations

Data Link Layer: Error Correction

- Hamming distance vs. the Hamming distance of a complete code
  - To reliably detect *d* errors, you need a distance *d* + 1 code
  - To correct *d* errors, you need a distance 2*d* + 1 code.
- Hamming codes (注意Hamming编码和Hamming距离是两个风马牛不相及的概念)

$$(m + r + 1) \leq 2^{r}$$

- Binary convolutional codes (虽然考试不一定会涉及但是在实际中却是应用非常普遍)

Data Link Layer: Error Detection

- Error-correcting codes are widely used on wireless links (因为无线链路error rate比较高)
- Over fiber or high-quality copper, the error rate is much lower, so **error detection** and **retransmission** is usually more efficient there for dealing with *the occasional error*.

- Error Detection
  - Parity
  - Checksum
    - The 16-bit Internet check
  - Cyclic Redundancy Checks (CRCs)
    - The pattern P is chosen to be one bit longer than the desired FCS.

T

P

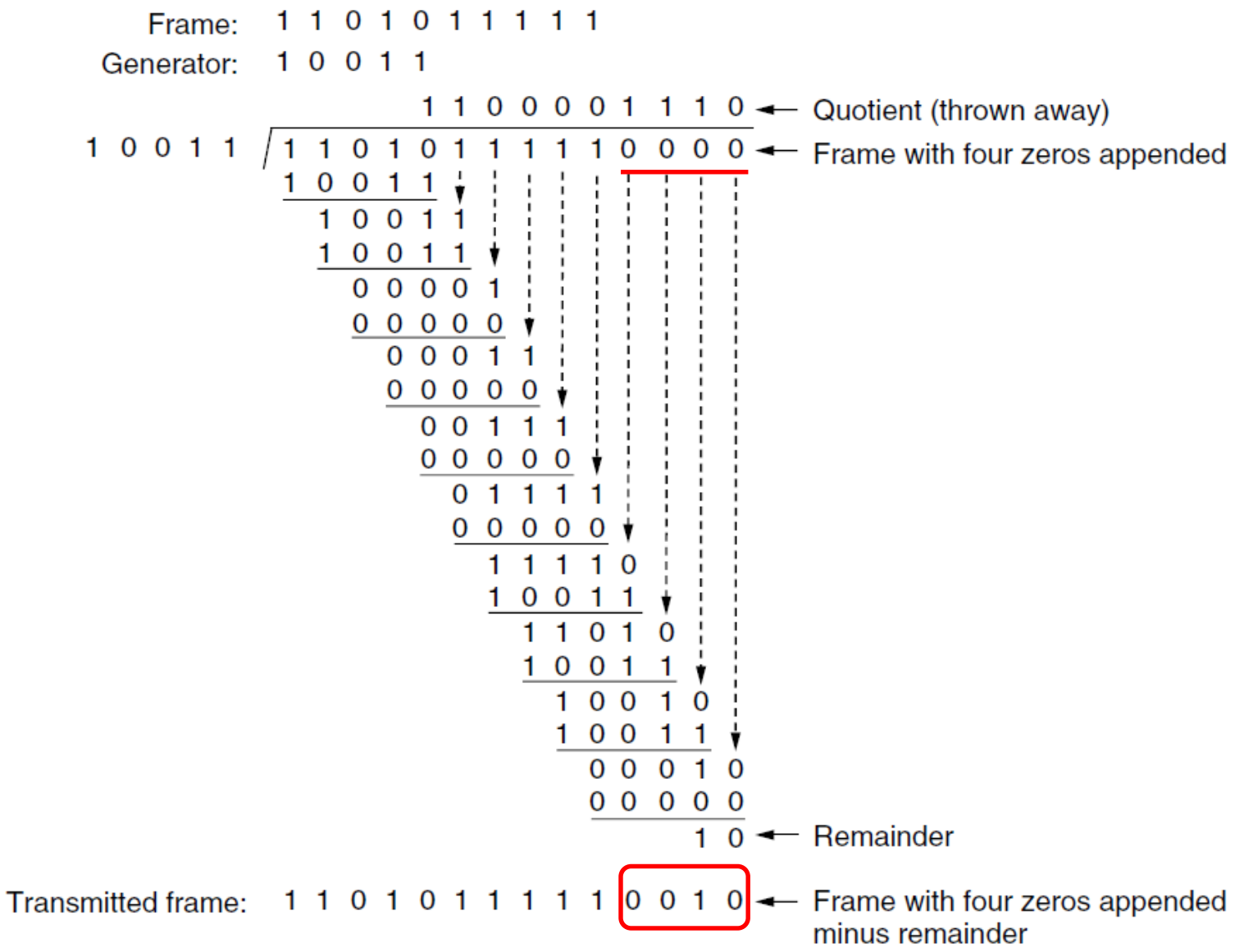


Figure 3-9. Example calculation of the CRC.

Think about if an error pattern is divisible by the generator P, what will happen?

Data Link Layer: Sliding Window Protocols

- A One-Bit Sliding Window Protocol
  - piggyback
- A Protocol Using Go Back N
  - The receive window of size 1.
  - There are MAX_SEQ+1 distinct sequence numbers(0, 1, 2, ... MAX_SEQ), but the sender window<=MAX_SEQ (Why?)
  - The go-back-n protocol works well *if errors are rare*, but if the line is poor it wastes a lot of bandwidth on retransmitted frames.
- A Protocol Using Selective Repeat
  - To allow the receiver to accept and buffer the frames following a damaged or lost one.
  - To ensure there is no overlap, the maximum sending window size should be at most half the range of the sequence numbers.
    - The sender window <= ((MAX_SEQ+1)/2) (Why?)
  - NAK

An Example

- Example: consider a 50-kbps satellite channel with a 500 msec round-trip propagation delay. Let us imagine trying to use protocol 4 (one-bit sliding window) to send 1000-bit frames via the satellite.
- At $t = 0$ the sender starts sending the first frame, at $t = 1000$ bit / (50×10^3) bps = 0.02sec = 20 msec the frame has been completely sent. Not until $t = 500/2 + 20 = 270$ msec has the frame fully arrived at the receiver, and not until $t = 520$ msec has the acknowledgement arrived back at the sender, under the best circumstances.
- But this means that the sender was blocked 500/520 or 96% of the time. In other words, only 4% of the available bandwidth was used.
- The problem described here can be viewed as a consequence of the rule requiring a sender to wait for an acknowledgement before sending another frame.
  - The protocol 4 is disastrous in terms of efficiency under the situation of **a long transit time**, **high bandwidth**, and **short frame length**.

An Example (cont.)

- Basically, the solution lies in allowing the sender to transmit up to *w* frames before blocking, instead of 1.
- How to find an appropriate value for *w* ?
  - 1) This capacity is determined by the bandwidth in bits/sec multiplied by the one-way transit time, or **the bandwidth-delay product** of the link. $50 \times 10^3 \times 250 \times 10^{-3} = 12.5 \times 10^3$ bits
  - 2) We can divide this quantity by the number of bits in a frame to express it as a number of frames. — $BD = 12.5 \times 10^3$ bits / 1000 bits/frame = 12.5 frames
  - 3) *w* should be set to $2BD + 1$. ($w = 26$ frames)

Twice the bandwidth-delay is the number of frames that can be outstanding if the sender continuously sends frames when the round-trip time to receive an acknowledgement is considered. The "+1" is because an acknowledgement frame will not be sent until after a complete frame is received.

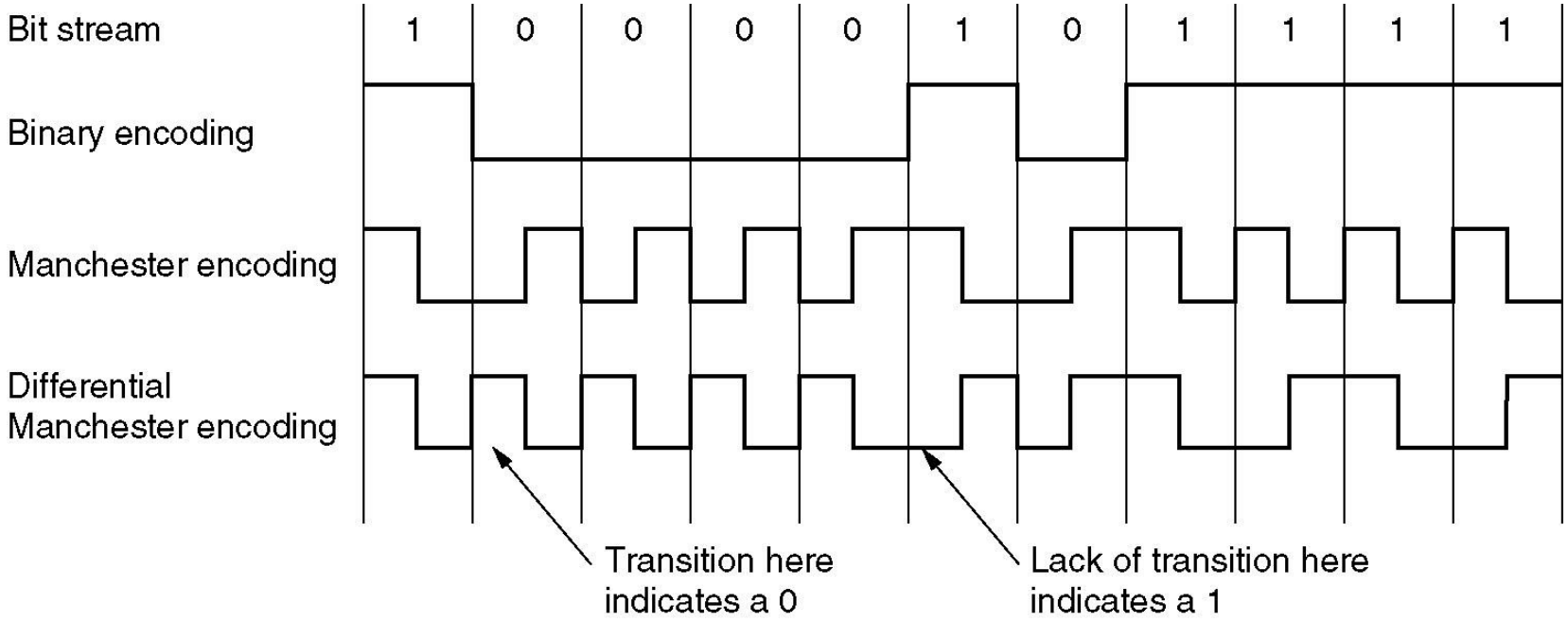
MAC SUB-LAYER

Protocols

- ALOHA
  - Pure ALOHA vs slotted ALOHA
- Carrier Sense Multiple Access Protocols
  - 1-persistent CSMA
  - Nonpersistent CSMA
  - *p*-persistent CSMA
  - CSMA/CD (Collision Detection)
- Collision Free Protocols
  - A Bit-Map Protocol
  - Token Passing
  - Binary Countdown
  - Limited-contention Protocols

Ethernet

- Two kinds of Ethernet exist:
  - Classical Ethernet (3 to 10 Mbps)
  - Switched Ethernet, in which devices called switches are used to connect different computers, runs at 100, 1000 and 10,000 Mbps, in forms called fast Ethernet, gigabit Ethernet, and 10 gigabit Ethernet.
  - Over the cables, information was sent using **the Manchester encoding**.



Ethernet Frame Structure

- **Preamble** of 8 bytes, each containing the bit pattern 10101010 (with the exception of the last byte, in which the last 2 bits are set to 11).
  - To allow the receiver's clock to synchronize with the sender's.
- The **48**-bit number of addresses
- The Ethernet requires that **valid frame must be at least 64 bytes long** *from destination address to checksum, including both*.

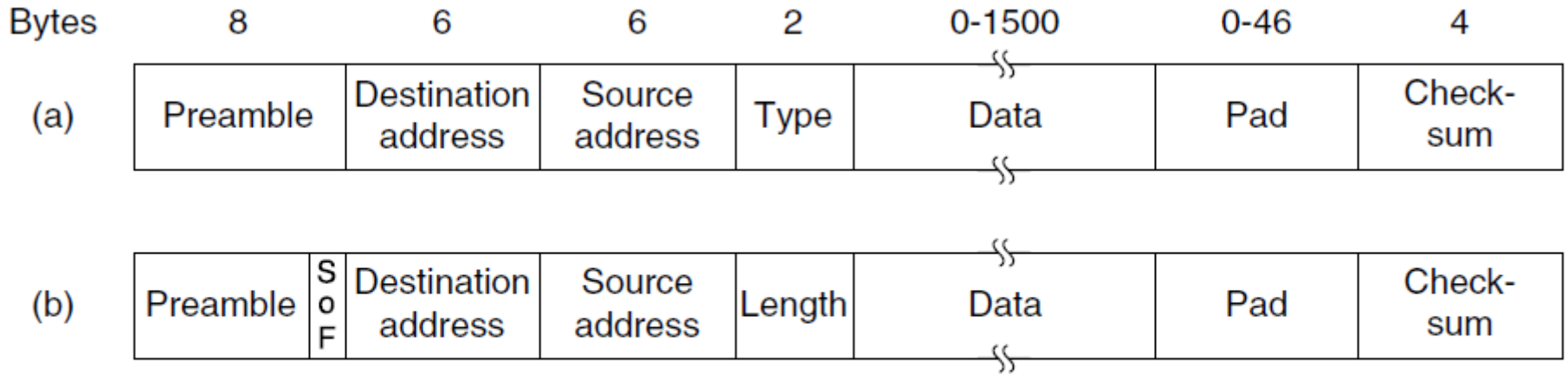


Figure 4-14. Frame formats. (a) Ethernet (DIX). (b) IEEE 802.3.

CSMA/CD with Binary Exponential Backoff

- Classic Ethernet uses the 1-persistent CSMA/CD algorithm.
  - A station senses the channel when it has a frame to send and send the frame as soon as the channel becomes idle.
- How the random interval is determined when a collision occurs?
  - After a collision, time is divided into discrete slots whose length is equal to the worst-case round-trip propagation time on the Ethernet (2τ).
    - To accommodate the longest path allowed by Ethernet, the slot time has been set to 512 bit times (one bit duration is 100 nsec), or 51.2μsec.
  - After i collisions when $i \leq 10$, a random number k between 0 and 2^i-1 slot times is chosen.
  - After 10 collisions have been reached, the randomization interval is frozen at a maximum of 1023 slots.
  - After 16 collisions, the controller throws in the towel and reports failure back to the computer.

802.11 WiFi

- **802.11 WiFi**
  - The **exposed** terminal and the **hidden** terminal problems
  - **MACA** (Multiple Access with Collision Avoidance, Karn, 1990): **A short handshake** before sending a frame, RTS (Request to Send) and CTS (Clear to Send)
  - 802.11 Frame Structure

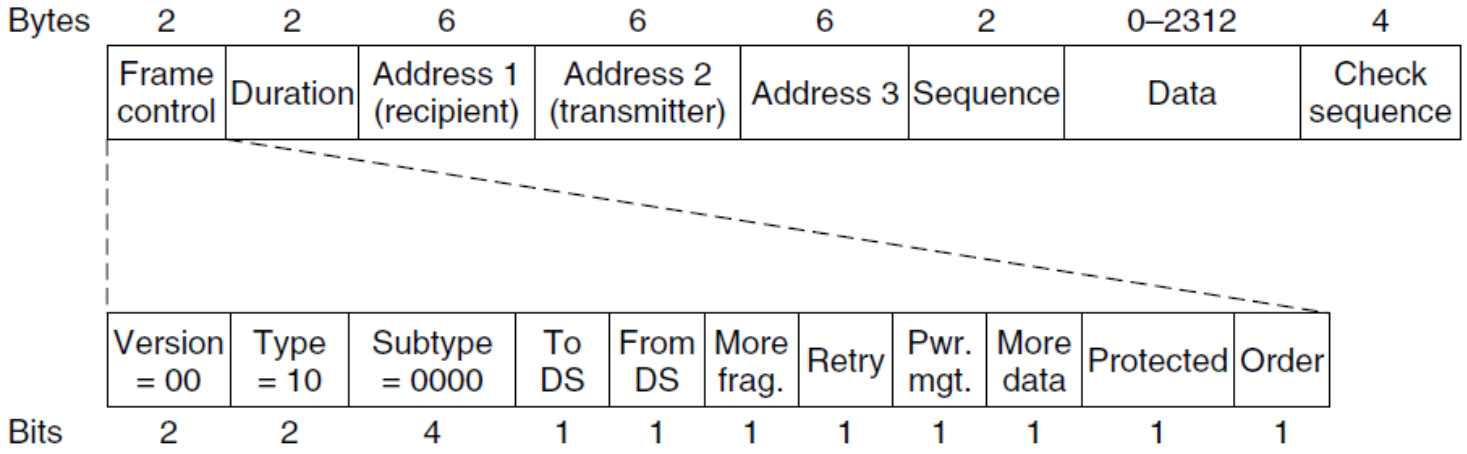


Figure 4-29. Format of the 802.11 data frame.

Bridge

- All of the stations attached to the same port on a bridge belong to the same collision domain, and this is different than the collision domain for other ports.
- Bridges can contribute to security: promiscuous mode

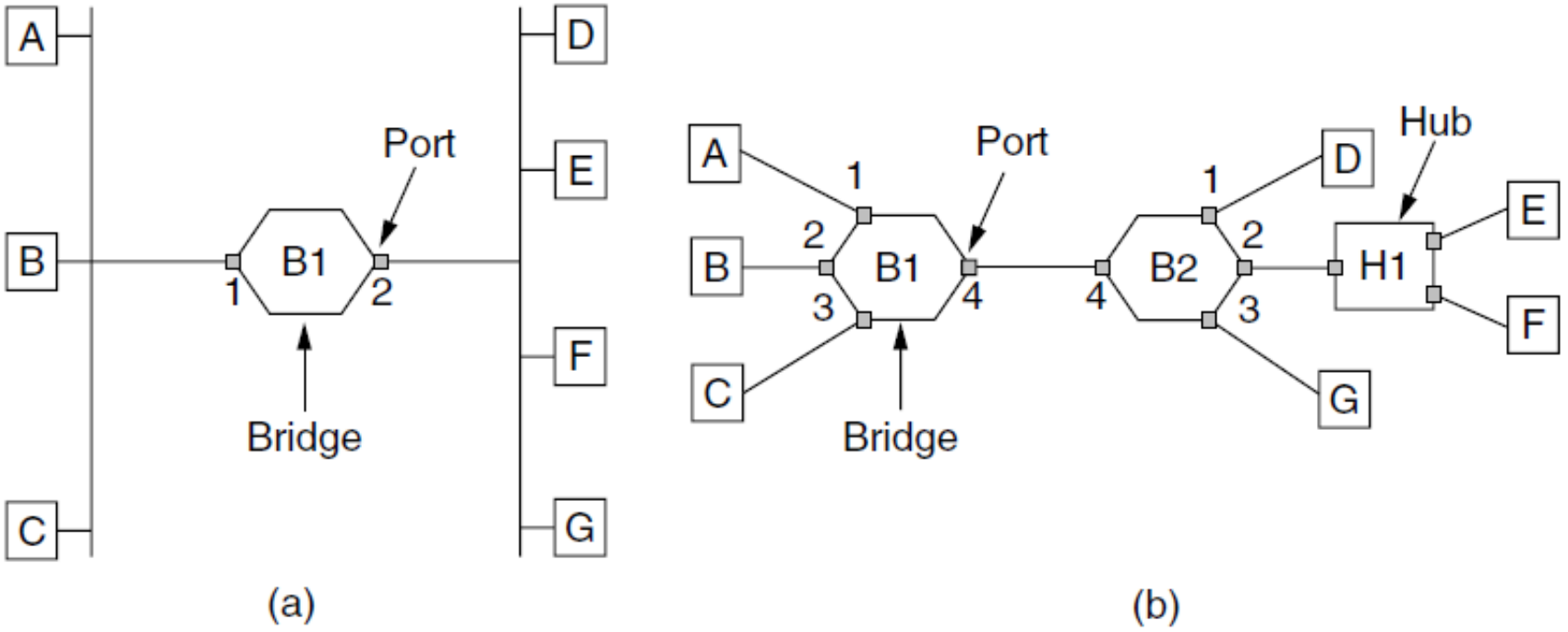


Figure 4-41. (a) Bridge connecting two multidrop LANs. (b) Bridges (and a hub) connecting seven point-to-point stations.

Learning Bridge

- The bridge must decide whether to forward or discard each frame, and, if the former, on which port to output the frame.
  - This decision is made by using the destination address.
- A simple way to implement this scheme is to have a big (hash) table. This table can list each possible destination and which output port it belongs to.
- When the bridges are first plugged in, all the hash tables are empty. None of the bridges know where any of destinations are, so they use **a flooding algorithm**.
  - Every incoming frame for an unknown destination is output on all the ports to which the bridge is connected except the one it arrived on.
  - Once a destination is known, frames destined for it are put only on the proper port.

Repeaters, Hubs, Bridges, Switches, Routers, and Gateways

- The key to understanding these devices is to realize that they operate in different layers
  - The layer matters because *different devices use different pieces of information to decide how to switch.*

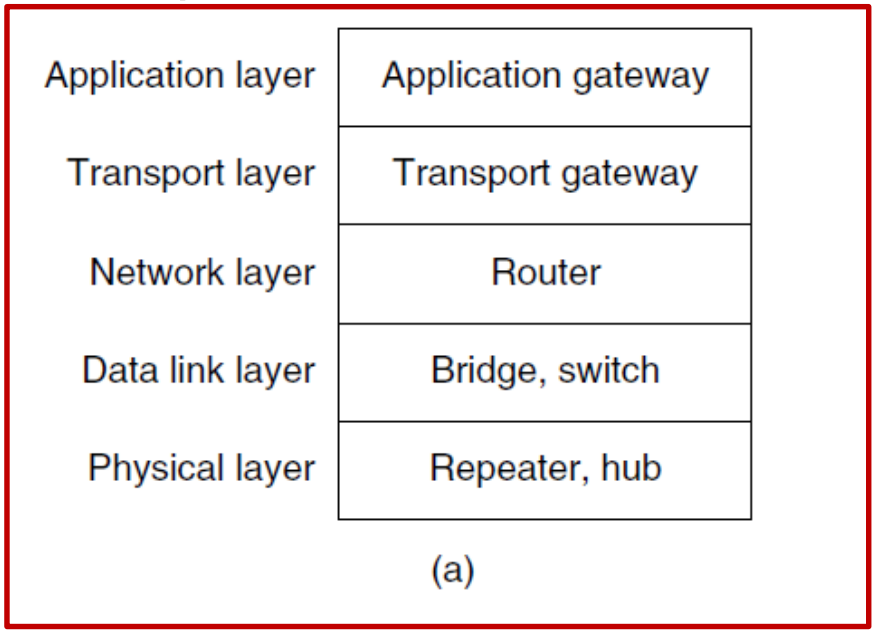


Figure 4-45. (a) Which device is in which layer. (b) Frames, packets, and headers.

Spanning Tree

- The solution to this difficulty is switches collectively find **a spanning tree** for the topology.
  - A spanning tree is a subset of links that is **a tree** (no loops) and reaches all switches.
  - There is a **unique** path from each source to each destination

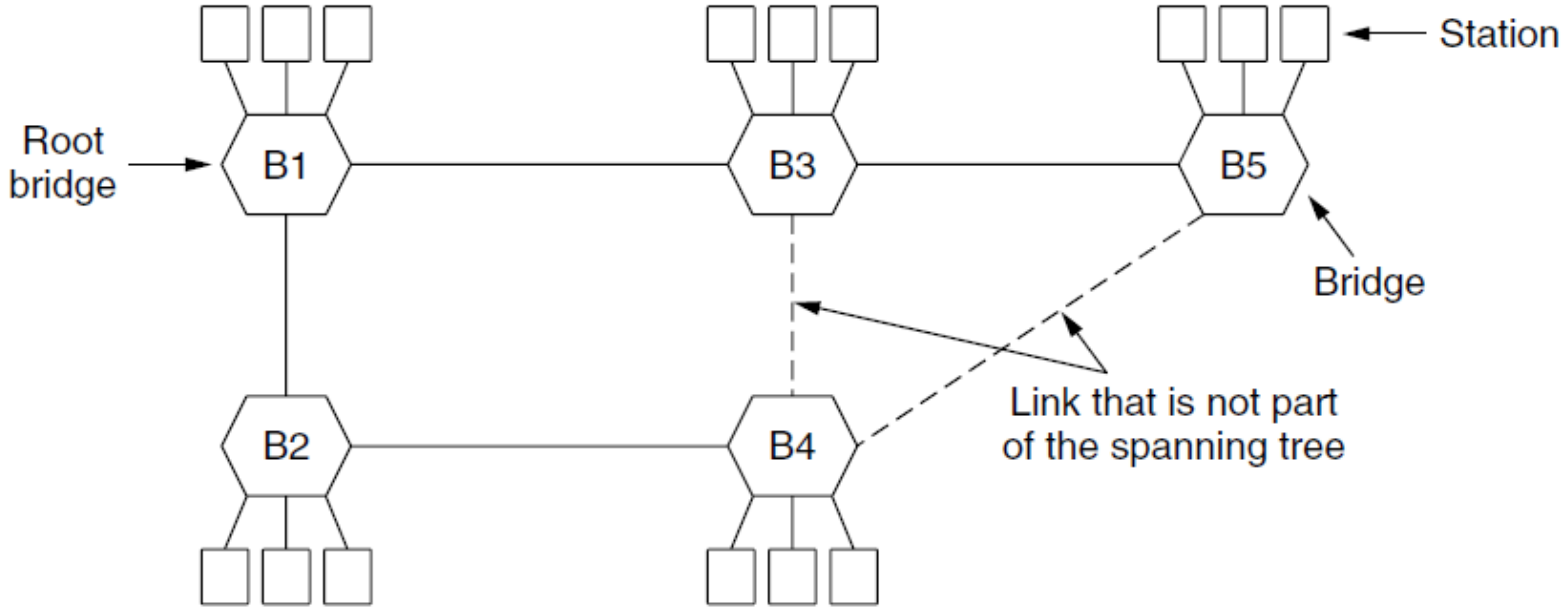


Figure 4-44. A spanning tree connecting five bridges. The dashed lines are links that are not part of the spanning tree.

Spanning Tree Algorithm (I)

- To build the spanning tree, the switches run a distributed algorithm.
- Each switch periodically broadcasts **a configuration message** out all of its ports to neighbors and processes the messages it receives from other bridges. These messages are **not** forwarded, since their purposes is to build the tree, which can then be used for forwarding.
  - 1. Select a root node (switch with the **lowest** address (MAC address))
  - 2. Grow the tree as shortest distances from the root (using the lowest address to break distance ties).
  - 3. Turn off the port for forwarding if they are not on the spanning tree.

Spanning Tree Algorithm (II)

- Details:
  - Each switch *initially* believes it is the root of the tree.
  - Each switch sends *periodic* updates to neighbors with: **its address**, **address of root**, and **distance (in hops) to root**.
  - Switches favor ports with shorter distance to lowest root.
    - To use lowest address to break distance tie.

Virtual VLANs

- Virtual LANs can **decouple** the logical topology from the physical topology. — To rewire buildings entirely *in software*.
  - Based on **VLAN-aware switches**.
  - Ports in each VLAN form a broadcast domain.
  - VLAN trunking: a special port on each switch is configured as a trunk port to interconnect the two VLAN switches.
  - **The trunk port** belongs to all VLANs, and frames sent to any VLAN are forward over the trunk link to the other switches.
    - switchport mode trunk
  - 802.1Q Frame Format

NETWORK LAYER

Two Important Network Layer Functions [2]

- **Forwarding** (the main function of a router)
  - Forwarding involves the transfer of a packet from an incoming link to an outgoing link within a single router.
  - Forwarding refers the router-local action.
- **Routing** (to build *the forwarding table* for each router)
  - Routing involves all of the network's routers, whose collective interactions via routing protocols determine the paths that packets take on their trips from source to destination node. The routing algorithm determines the values that are inserted into the routers' forwarding tables.
  - Routing refers to the network-wide process.
    - Centralized or decentralized.

Virtual-Circuit vs. Datagram Networks

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

Figure 5-4. Comparison of datagram and virtual-circuit networks.

Classification of Routing Algorithms [2]

- We can classify routing algorithms into global routing algorithms and decentralized algorithms.
  - **Global** routing algorithms compute the least-cost path between a source and destination using complete, global knowledge about the network.
    - **Link-state** (LS) algorithms
      - Dijkstra Algorithm
  - In **decentralized** routing algorithms, the calculation of the least-cost path is carried out in an iterative, distributed manner.
    - **Distance-vector** (DV) algorithms

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

      - The count-to-infinite problem

Link State Routing *vs.* Distance Vector Routing

- Message Complexity
  - LS: with n nodes, E Links, $O(nE)$ messages sent
  - DV: exchange between neighbors only

- Speed of Convergence
  - LS: $O(n^2)$ algorithm, requires $O(nE)$ messages
    - may have oscillations
  - DV: convergence time varies
    - count-to-infinite problem

- Robustness: what happens if router malfunctions?

  LS:
  - node can advertise incorrect link cost
  - each node computes only its own table

  DV:
  - DV node can advertise incorrect path cost
  - each node's table used by others
    - errors propagate through network

The Network Layer of the Internet [2]

- The network layer of the internet has three main components:
  - The **IP** protocol
  - The Internet routing protocols (including **RIP**, **OSPF** and **BGP**)
  - The Internet control protocols (including **ICMP**, **DHCP**, **ARP**)

The IPv4 Datagram

- The header has a 20-byte fixed part and a variable-length optional part.
- The bits are transmitted from left to right and top to bottom. This is "big-endian" network byte order.

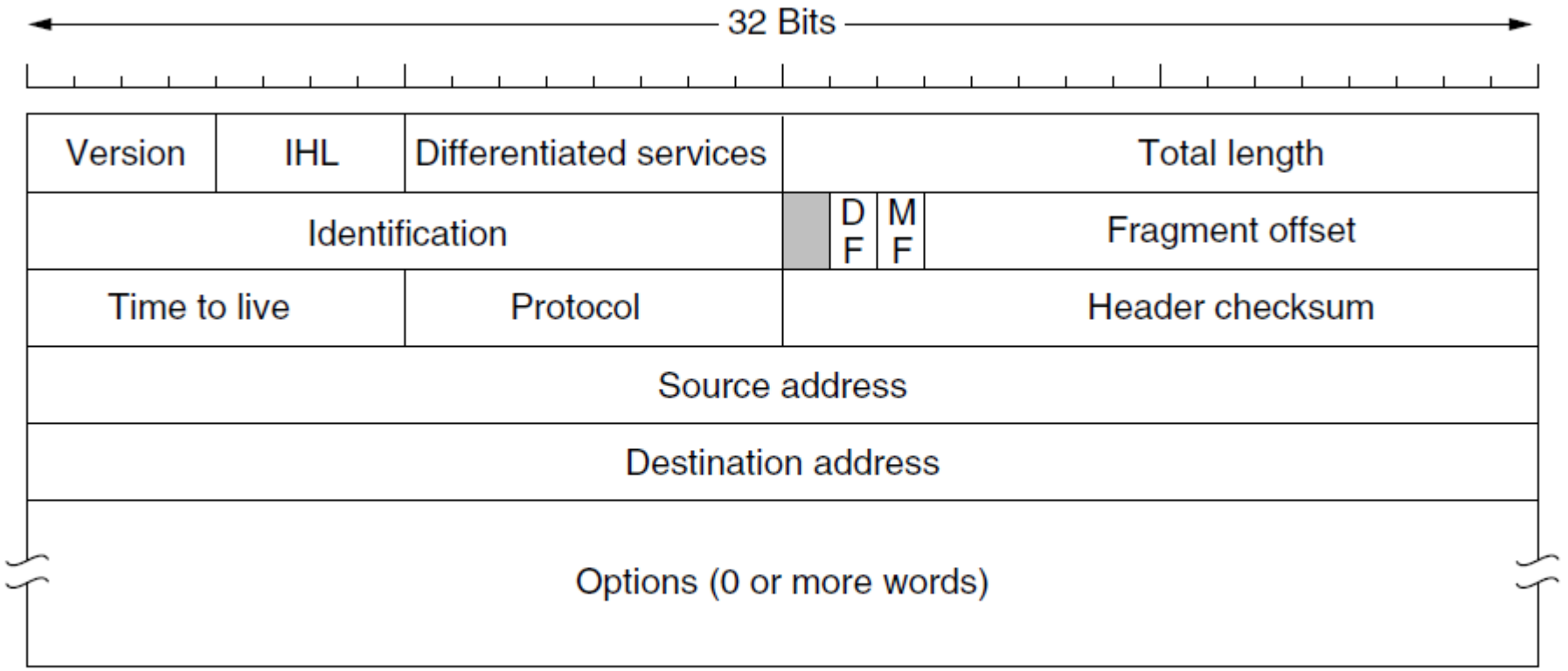


Figure 5-46. The IPv4 (Internet Protocol) header.

IPv4 Addressing (II)

- IP addresses are written in dotted decimal notation.
  - In this format, each of the 4 bytes is written in decimal, from 0 to 255.
  - Example: 128.208.2.151
- IP addresses can also be expressed in hexadecimal
  - Example: 128:208.2.151 = 80D00297
- Addresses are allocated in blocks called **prefixes**.
  - Addresses in an L-bit prefix have the same top L bits
  - There are 2^{32-L} addresses aligned on 2^{32-L} boundary.

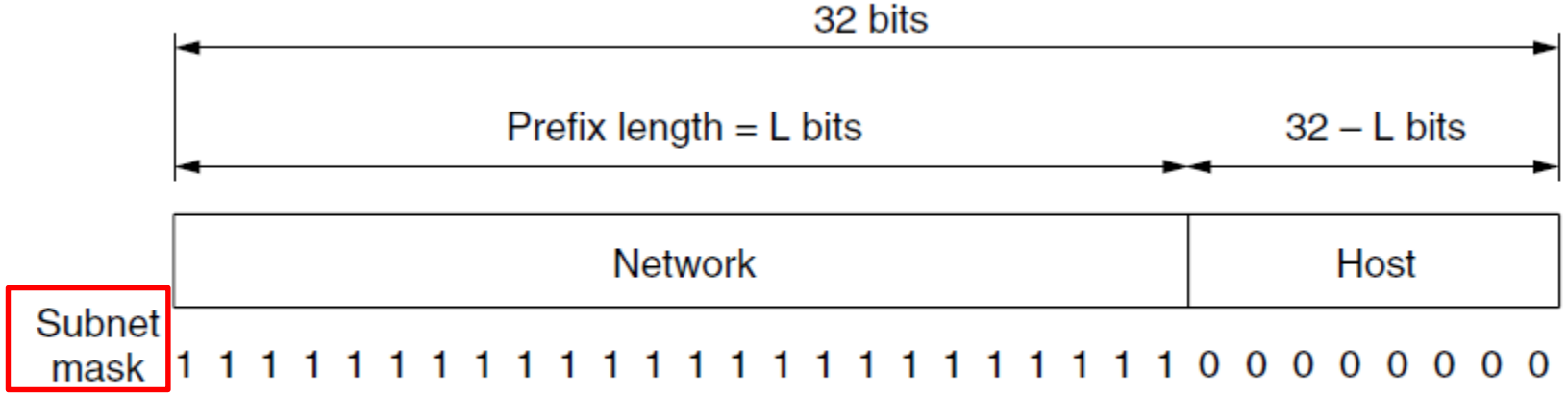


Figure 5-48. An IP prefix and a subnet mask.

IP Prefixes

- Written in “IP address/length” notation
  - Address is lowest address in the prefix, length is prefix bits. The /N sometimes known as a subnet mask.
    - /24 → The subnet mask is 255.255.255.0
  - E.g., 128.13.0.0/16 is 128.13.0.0 to 128.13.255.255
  - So a/24 is 256 addresses, and a/32 is one address.

- The key advantage of prefixes is that routers can forward packets *based on only the network portion of the address*, as long as each of the networks has a unique address block.
  - More specific prefix has longer prefix, hence a smaller number of IP addresses.
  - Less specific prefix has shorter prefix, hence a larger number of IP addresses.

IP Address Classes - Historical

- Before CIDR (Classless InterDomain Routing) was adopted, the network portions of an IP address were constrained to be 8, 16, or 24 bits in length, and addressing scheme known as **classful addressing**.

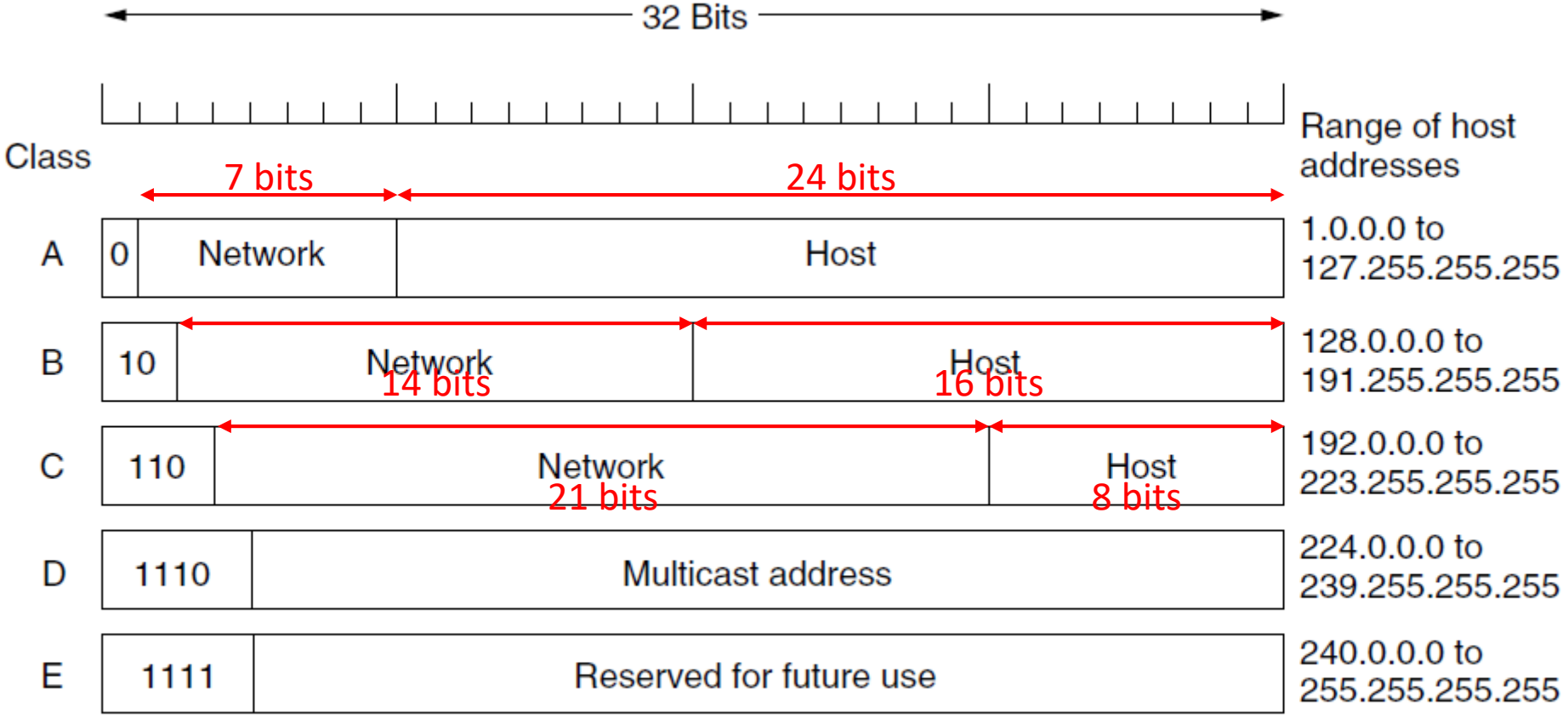


Figure 5-53. IP address formats.

CIDR — Classless InterDomain Routing

- Even if blocks of IP addresses are allocated so that the addresses are used efficiently, there is still a problem that remains: **routing table explosion**. [RFC 4632]
- There is something we can do to reduce routing table sizes — By adjusting the size of IP prefixes
- **Subnetting**: split IP prefixes
- **Route aggregation**: combine multiple small prefixes into a single larger prefix.
- The design work of subnetting and route aggregation is called CIDR (Classless InterDomain Routing).

Special IP Addresses

- The IP address 0.0.0.0, the lowest address, is used by hosts when they are being booted. It means “this network” or “this host”.
- The IP address 255.255.255.255, the highest address, is used to mean all hosts on the indicated network. It allows broadcasting on the local network, typically a LAN.
- The IP address 127.0.0.1 (本机地址), and 127.xx.yy.zz are reserved for loopback testing.

The IPV6 Header (I)

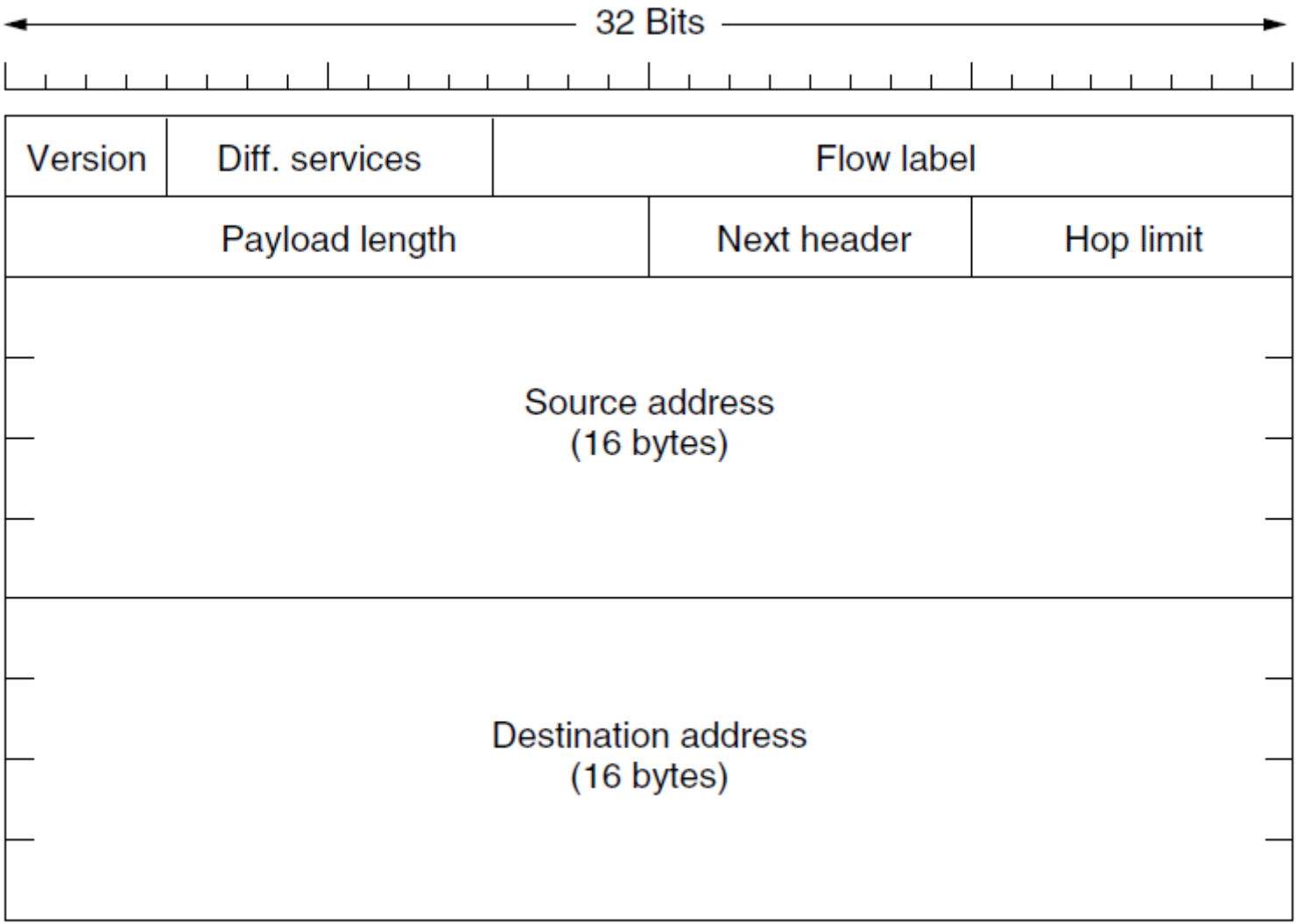


Figure 5-56. The IPv6 fixed header (required).

The IPv6 Address

- IPv6 addresses are written as eight groups of four hexadecimal digits with colons between the groups

  8000:0000:0000:0000:0123:4567:89AB:CDEF

- Since many addresses will have many zeros inside them, three optimization have been authorized:
  - Leading zeros with a group can be omitted, so 0123 can be written as 123.
  - One or more groups of 16 zero bits can be replaced by a pair of colons

    8000::123:4567:89AB:CDEF

  - IPv4 addresses can be written as a pair of colons and an old dotted decimal number ::192.31.20.46

Packet Fragmentation

- The maximum payloads of different networks
  - Ethernet 1500 bytes
  - 802.11 2272 bytes
  - IP 65,515 bytes

- Solutions
  - 1. To make sure the packet fragmentation does not occur in the 1st place.
    - Path MTU (Path Maximum Transmission Unit)
  - 2. To break up packets into fragments, sending each fragment as a separate network layer packet.
    - Two opposing strategies exist for recombining the fragments back into the original packet: **transparent fragmentation** or **nontransparent fragmentation**.

IP Fragment Example [2]

Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 = 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980–1,480–1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 = 2,960)	flag = 0 (meaning this is the last fragment)

- 分片偏移就是某片在原分组的相对位置，以8个字节为偏移单位。这就是说，每个分片的长度一定是8字节（64位）的整数倍。
- 每个分片都要加上IP头部。
- MF = 0 means this the last fragment (MF is a flag bit in a IP datagram, MF — More Fragments)

IP addresses are scarce!

- NAT
- DHCP

Internet Control Message Protocol (**ICMP**) [2]

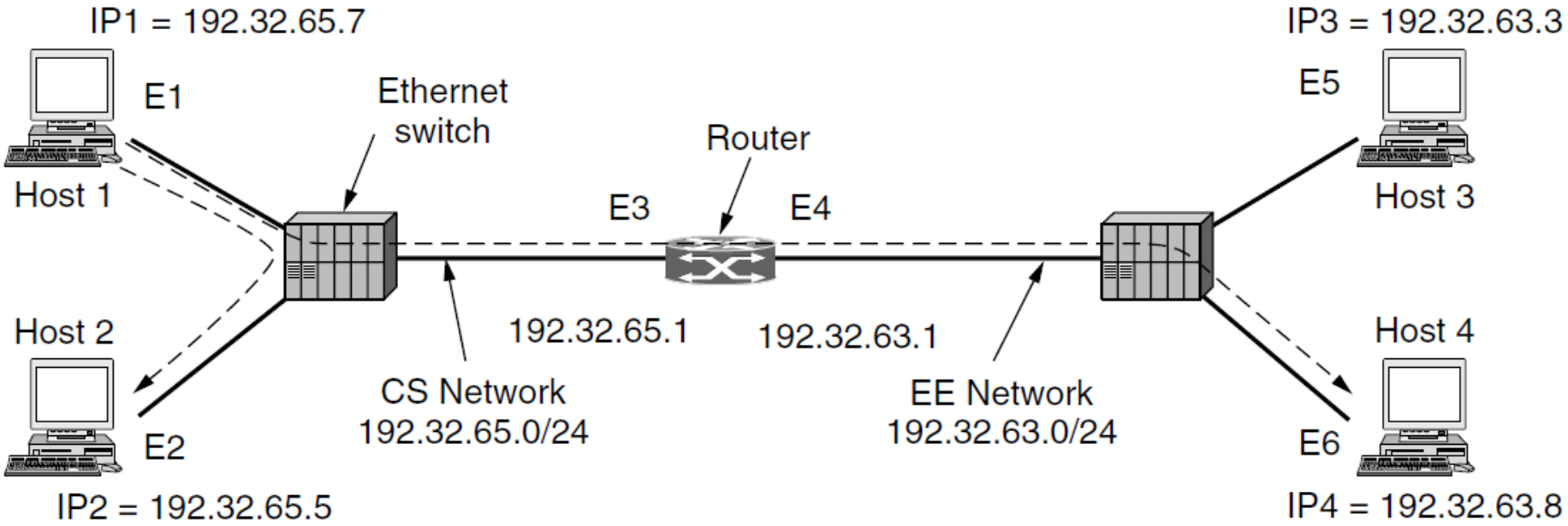
- ICMP is specified in RFC 792.
- The most typical use of ICMP is for **error reporting**.
  - For example, when running a Telnet, FTP, or HTTP session, you may have encountered an error message such as "*Destination network unreachable*".
- ICMP is often considered part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams.
- ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram.
- Using ICMP
  - **Ping**
  - **Tracert (traceroute)**

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Figure 4.23 ♦ ICMP message types

ARP (The Address Resolution Protocol)

- Although every machine on the Internet has one or more IP addresses, these addresses are not sufficient for sending packets.
  - Data link layer NICs (Network Interface Cards) such as Ethernet cards do not understand Internet addresses.
  - The NICs send and receive frames based on 48-bit **Ethernet addresses** (the link layer addresses, that is **MAC addresses**).
- Now the question is: how do IP addresses get mapped onto data link layer addresses, such as Ethernet?
  - For the Internet, this is the job of the **Address Resolution Protocol** (**ARP**) [RFC826].
  - The purpose of the ARP query packet is to query all the other nodes *on the subnet* to determine the MAC address corresponding to the IP address that is being resolved.



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

Figure 5-61. Two switched Ethernet LANs joined by a router.

Observe that the Ethernet address change with the frame on each network while the IP addresses remain constant (because they indicate the endpoints across all of the interconnected networks).

ARP vs. DNS

- ARP vs. DNS
  - ARP resolves an IP address to a MAC address only for nodes on the same subnet.
  - DNS resolves host names to IP addresses for hosts anywhere in the Internet.
- ARP is probably best considered a protocol that straddles the boundary *between the link and network layers*.

Label Switching and MPLS

- So far, we have focused exclusively on packets as **datagrams** that are forwarded by IP routers.
- MPLS (Multiprotocol Label Switching) is perilously close to circuit switching.
  - To improve *the forwarding speed* of IP routers by adopting a key concept from the world of virtual-circuit networks: **a fixed-length label**.
  - The goal was not to abandon the destination-based IP datagram-forwarding infrastructure for one based on fixed-length labels and virtual circuits, but to augment it by selectively labeling datagrams and allowing routers to forward datagrams based on fixed-length labels (rather than destination IP addresses) when possible. (因为MPLS是固定长度的标签来routing，所以它比基于IP地址要快很多，实际应用系统应用比较普遍。但不定是考试重点)
  - The MPLS protocol [RFC 3031, RFC 3032]

Intra-AS Routing in the Internet: RIP

- **RIP** (Routing Information Protocol)
  - Based on **distance vector routing algorithm** [RFC 1058, RFC 2453]
  - Metric: **hop** count (1: directly connected, **16:infinity**)
  - Works well in small systems, but less well as networks get larger.
  - Supports networks with diameter ≤ 15
  - Suffers from **the count-to-infinity problem**
  - Messages carried in **UDP** datagrams using port number **520**. (~ Lab 5)

Intra-AS Routing in the Internet: OSPF

- **OSPF** (Open Shortest Path First)
  - OSPF is **link-state** protocol that uses **flooding** of link state information and a Dijkstra least-cost path algorithm. [RFC2328]
  - Metric: default based on bandwidth, the larger the bandwidth, the lower the cost.
  - OSPF messages that are carried directly by **IP**, with an upper-layer protocol of **89** for OSPF.
  - OSPF allows an AS to be divided into **numbered areas**.
    - Every AS has **a backbone area**, called **area 0**.
    - A star configuration on OSPF, with the backbone being the hub and other areas being spokes.
    - Each router that is connected to two or more areas is called **an area border router**.
    - One router is elected as **the designed router**. It acts as **the single node** that represents the LAN. **A backup designed router** is always kept up to date to ease the transition should the primary designed router crash and need to be replaced immediately. (~ Lab 5)

Inter-AS Routing in the Internet: BGP

- The BGP (Border Gateway Protocol) is the de facto standard inter-AS routing protocol in today's Internet. [RFC 4271; RFC 4274; RFC 4276]
  - BGP is a form of **distance vector routing protocol**
  - Metric: *based on policies* rather than on minimum distance
  - BGP chooses a path to follow at the AS level and OSPF chooses paths within each of the ASs.
  - In BGP, pairs of routers exchange routing information over **semipermanent TCP connections** using port **179**.
    - A mesh of TCP connections within each AS.
    - In BGP, **destinations** are **not** hosts but instead are **CIDRized prefixes**, with each prefix representing a subnet or a collection of subnets.

TRANSPORT LAYER

Transit Units of Different Layers

- Transport layer: segment or TPDU (Transport Protocol Data Unit)
- Network layer: packet
- Data link layer: frame
- Physical layer: bit

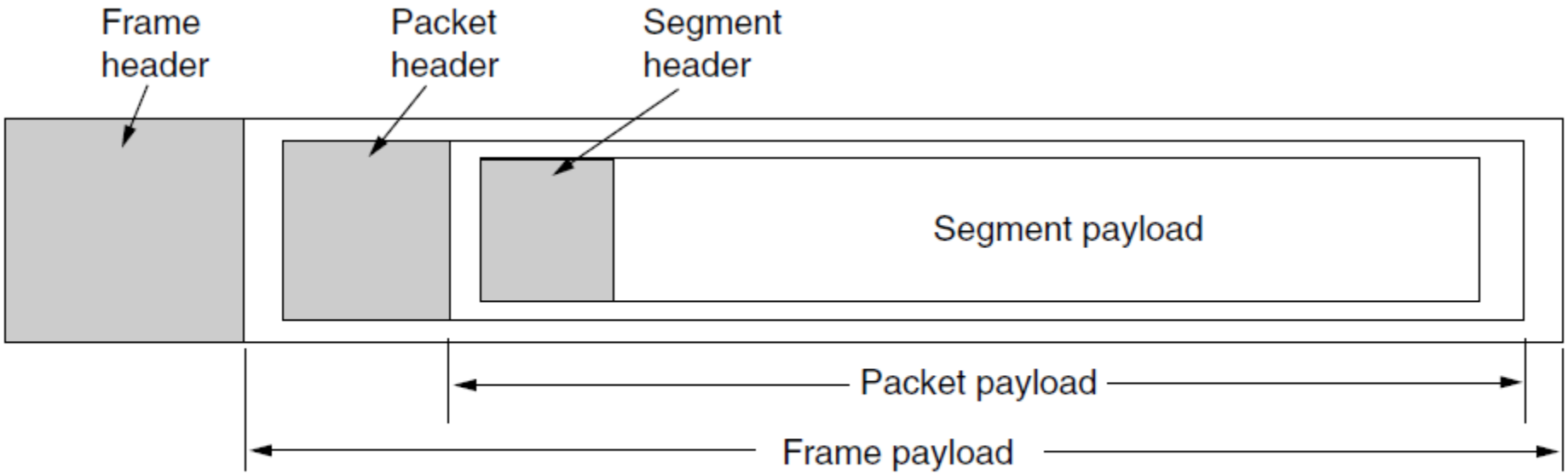


Figure 6-3. Nesting of segments, packets, and frames.

The Internet Transport Protocols

- The Internet has *two main protocols* in the transport layer
  - **UDP** (**User Datagram Protocol**, *connectionless* protocol): It does nothing beyond sending packets between applications. It typically runs in the operating system.
  - **TCP** (*connection-oriented* protocol): It does almost everything. It makes connections and adds reliability with retransmission, along with flow control and congestion control.

The Internet Transport Protocols: UDP

- What UDP does **not** do
  - Flow control, congestion control, or retransmission upon receipt of a bad segment.
- What UDP does **do**
  - To provide an interface to the IP protocol with the added feature of demultiplexing multiple processes using the ports.
  - Optional end-to-end error detection (~ checksum)
- Which application uses the UDP protocol
  - **DNS** (Domain Name System, Chapter 7)

The Internet Transport Protocols: UDP

- UDP: connectionless transport protocol
  - RFC768
  - UDP header (**8** bytes)

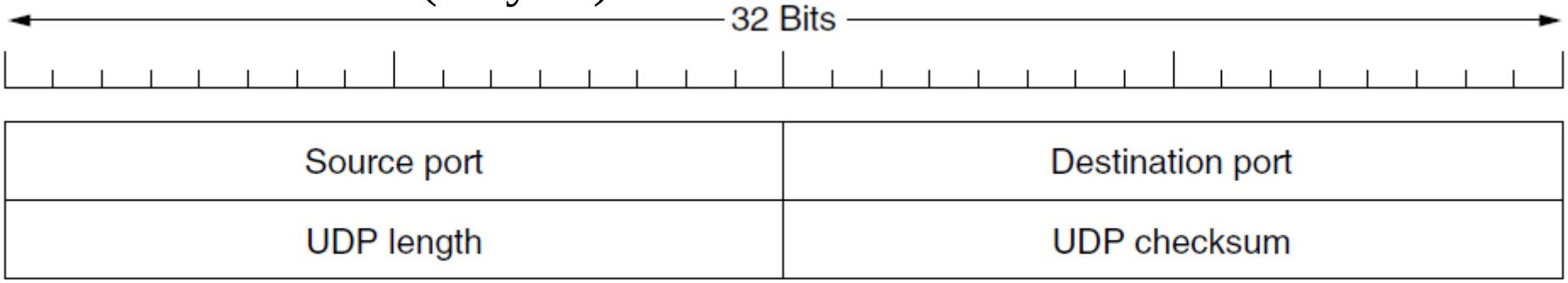


Figure 6-27. The UDP header.

- **The two ports** serve to identify the endpoints within the source and destination machines.
  - With these two ports, it delivers the embedded segment to the correct application.
  - The source port is primarily needed when a reply must be sent back to the source. By copying the Source Port field from the incoming segment into the Destination Port field of the outgoing segment.

Real-Time Transport Protocols

- **RTP** (Real-time Transport Protocols)
  - RFC3550
  - It is **a transport protocol** but just happens to be implemented in the application layer.
- Two aspects of real-time transport
  - The RTP protocol for transporting audio and video data in packets
  - How the receiver plays out the audio and video at the right time?

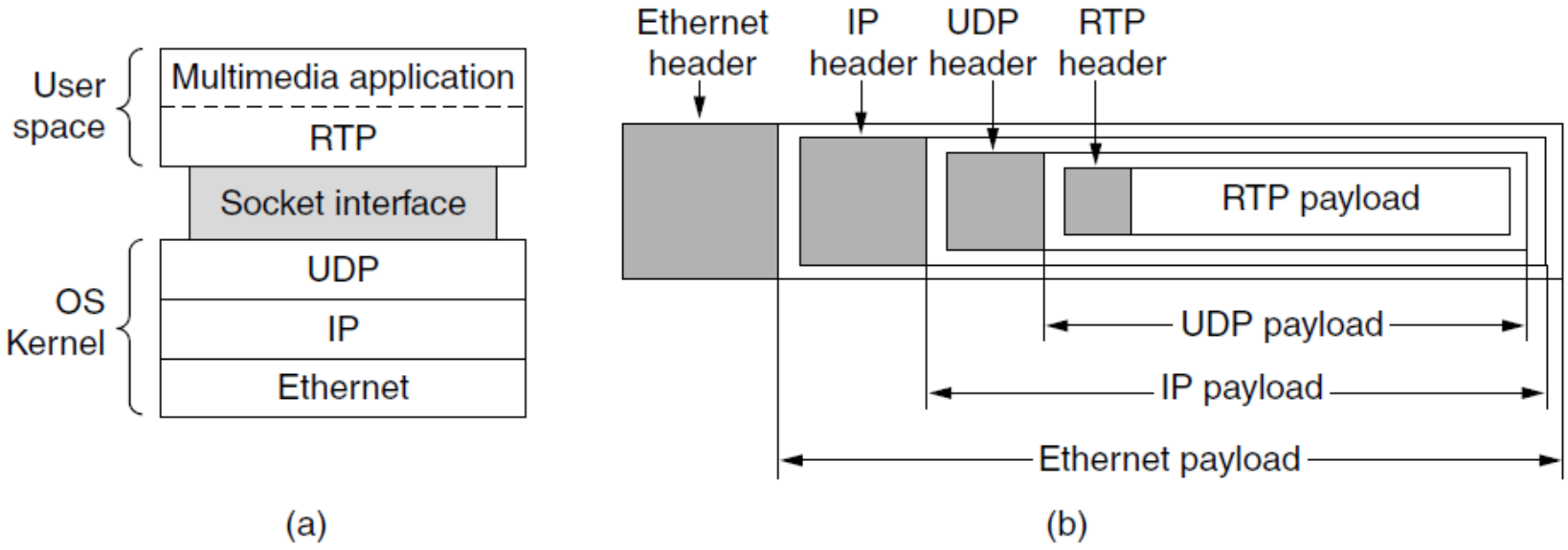


Figure 6-30. (a) The position of RTP in the protocol stack. (b) Packet nesting.

The Internet Transport Protocols: TCP

- TCP (Transmission Control Protocol) was designed to provide a reliable end-to-end byte stream over an unreliable internetwork.
- An internetwork may have wildly different topologies, bandwidths, delays, packet sizes, and other parameters in different parts.
- TCP was designed to dynamically adapt to properties of the internetwork and to be robust in the face of many kinds of failures.
  - RFC793, RFC793+, RFC1122 (clarifications and bug fixes), RFC1323 (extensions for high-performance), RFC2018 (selective acknowledgement), RFC 2581 (congestion control), RFC2873 (repurposing of header fields for quality of service), RFC2988 (improved retransmission timers), RFC 3168 (explicit congestion control)

The Internet Transport Protocols: TCP

- All TCP connections are **full duplex** and **point-to-point**.
  - TCP does **not** support broadcasting or multicasting
- A TCP connection is **a byte stream**, not a message stream.
  - When an application passes data to TCP, TCP may send it immediately or buffer it.
    - Force to send data immediately (PUSH flag)
    - Not only to send data immediately, but to process data immediately (URGENT flag)
- The basic protocol used by TCP entities is **the sliding window protocol** with dynamic window size.
  - When a sender transmits a segment, it starts a **timer**.
  - When the segment arrives at the destination, the receiving TCP entity sends back a segment (with data if any exist, and otherwise without) bearing **an ACK number equal to the next sequence number it expects to receive** and **the remaining window size**.
  - If the sender's timer goes off before the ACK is received, then the sender retransmits the segment.

The TCP Segment Header

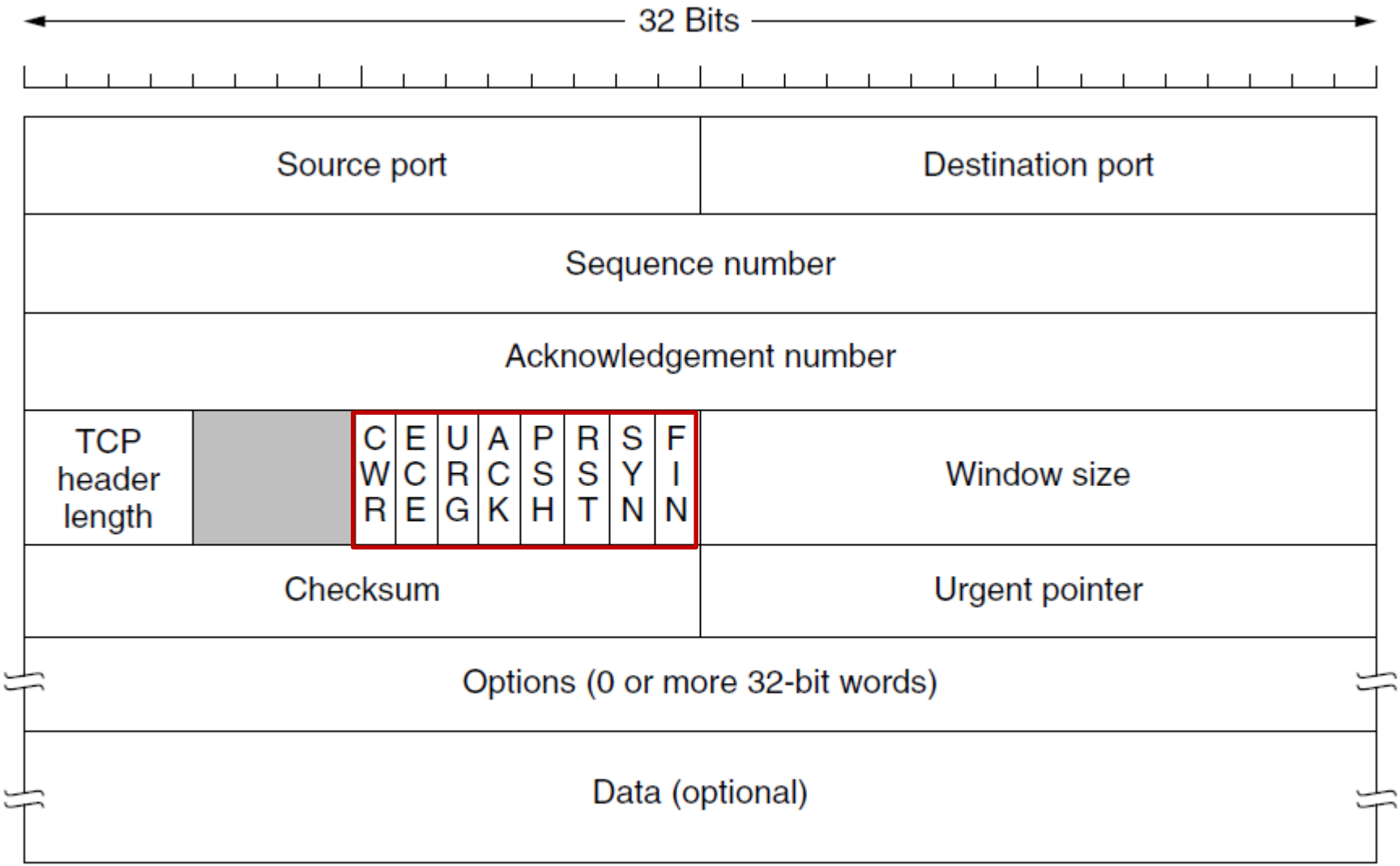
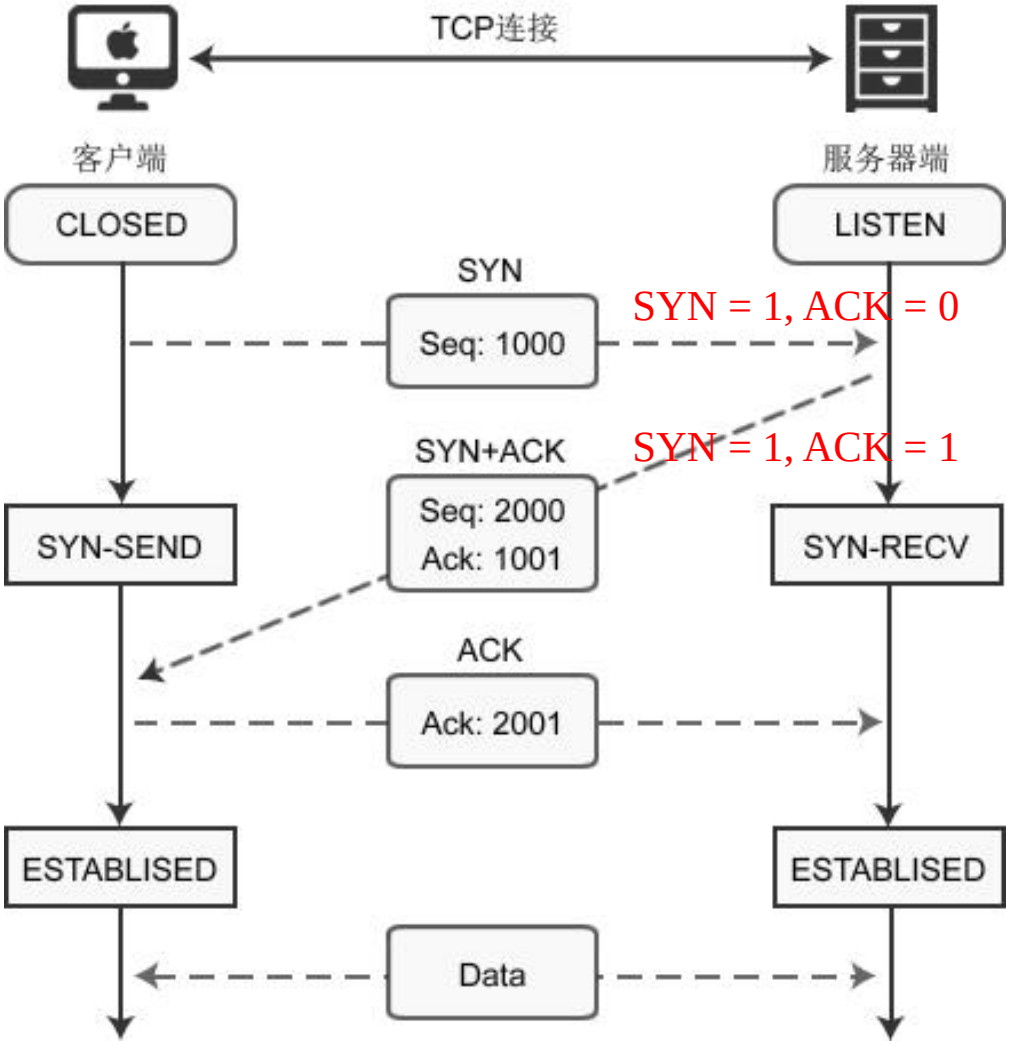


Figure 6-36. The TCP header.

TCP Connection Establishment: Three Way Handshake

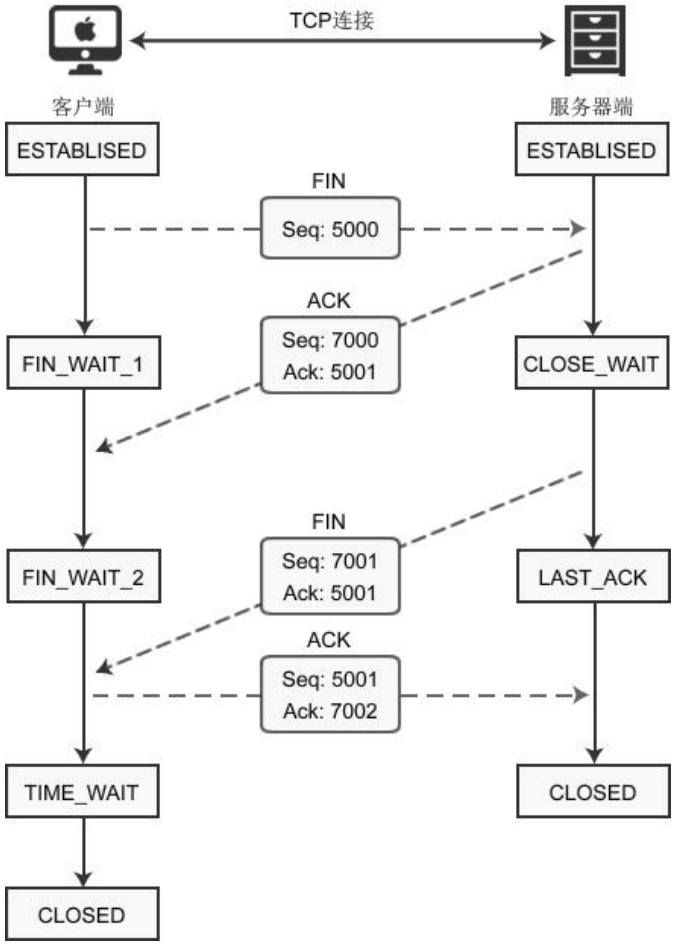
- In any TCP segment,
  - If SYN bit = 1 and ACK bit = 0, then it must be the request segment.
  - If SYN bit = 1 and ACK bit = 1, then it must be the reply segment.
  - If SYN bit = 0 and ACK bit = 1, then it can be the pure ACK or segment meant for data transfer.
  - If SYN bit = 0 and ACK bit = 0, then this combination is not possible.

TCP三次握手



三次握手的关键是要确认对方收到了自己的数据包，这个目标就是通过“确认号（ACK）”字段实现的。计算机会记录下自己发送的数据包序号 Seq，待收到对方的数据包后，检测“确认号（ACK）”字段，看ACK = Seq + 1是否成立，如果成立说明对方正确收到了自己的数据包。

TCP四次握手断开连接



注意：服务器收到请求后并不是立即断开连接，而是先向客户端发送“确认包”，告诉它我知道了，我需要准备一下才能断开连接。

TCP Timer Management

- TCP uses multiple timers (at least conceptually) to do its work.
  - The RTO (Retransmission TimeOut)
    - How long should the RTO be ? This problem is much more difficult in the transport layer than in data link protocols such as 802.11.
    - **Jacobson Algorithm** (1988): α×历史估计值 + (1 – α)×当前估计值
  - The Persistence timer is designed to prevent the following **deadlock**.
  - The Keepalive timer may go off to cause one side to check whether the other side is still there.
  - The TIME WAIT timer runs for **twice the maximum packet lifetime** to make sure that when a connection is closed, all packets created by it have died off.

TCP Sliding Window

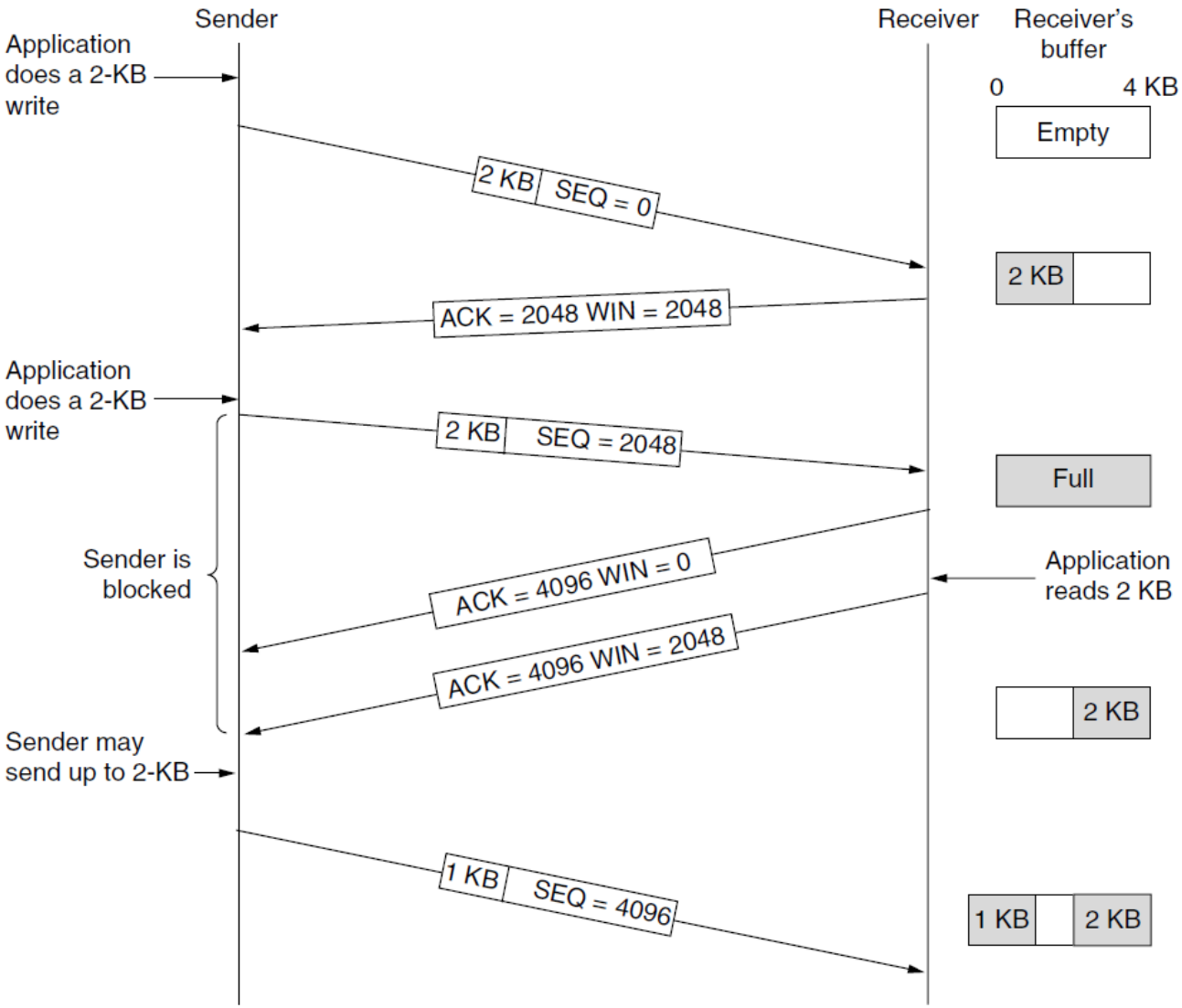


Figure 6-40. Window management in TCP.

ACK = 2048指下个期待的字节号 (It indicates the sequence number of the next data byte that receiver **expects** to receive from the sender)

TCP Sliding Window

- When the window is 0, the sender may not normally send segments, with two exceptions.
  - 1) Urgent data may be sent, for example, to allow the user to kill the process running on the remote machine.
  - 2) The sender may send a 1-byte segment to force the receiver to re-announce the next byte expected and the window size. This packet is called a **window probe**.
    - The TCP standard explicitly provides this option to prevent deadlock if a window update ever gets lost.

TCP Congestion Control

- TCP congestion control is based on a **AIMD** (**Additive Increase Multiplicative Decrease**) control law using a window and with **packet loss** as the binary signal.
- TCP maintains *a congestion window* and *a flow control window*
  - **The congestion window** whose size is the number of bytes the sender may have in the network at any time.
    - The corresponding data rate is the window size divided by the round-trip time of the connection
  - **The flow control window** which specifies the number of bytes that the receiver can buffer.
  - Both windows are tracked in parallel, and the number of bytes that may be sent is the smaller of the two windows.
  - TCP will stop sending data if either the congestion or the flow control window is temporarily full.
  - All the Internet TCP algorithms *assume* that lost packets are caused by congestion and monitor timeouts.

TCP Congestion Control

- **The key observation** is that the acknowledgements return to the sender at about the rate that packets can be sent over *the slowest link* in the path. This is precisely the rate that the sender wants to use.
- This timing is known as **an ack clock**. It is an essential part of TCP.
  - By using an ack clock, TCP smoothes out traffic and avoids unnecessary queues at routers.

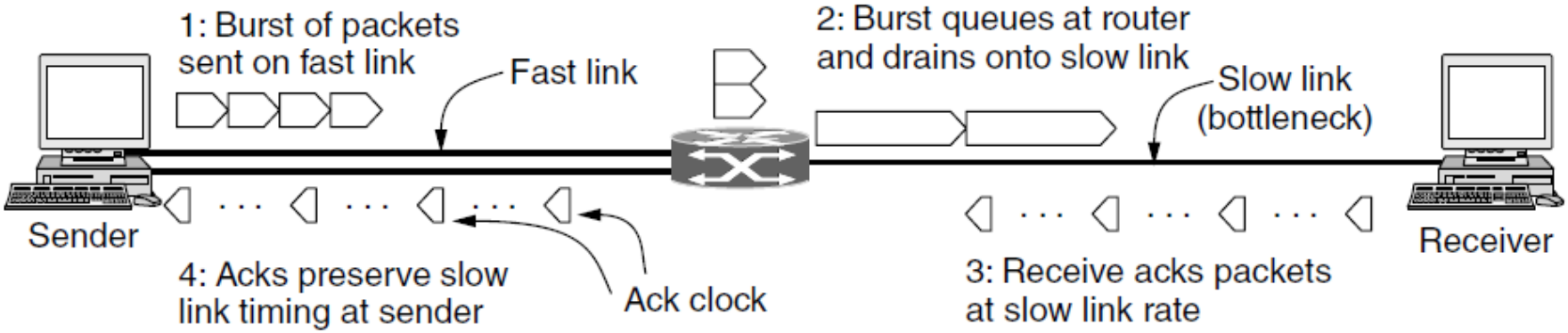
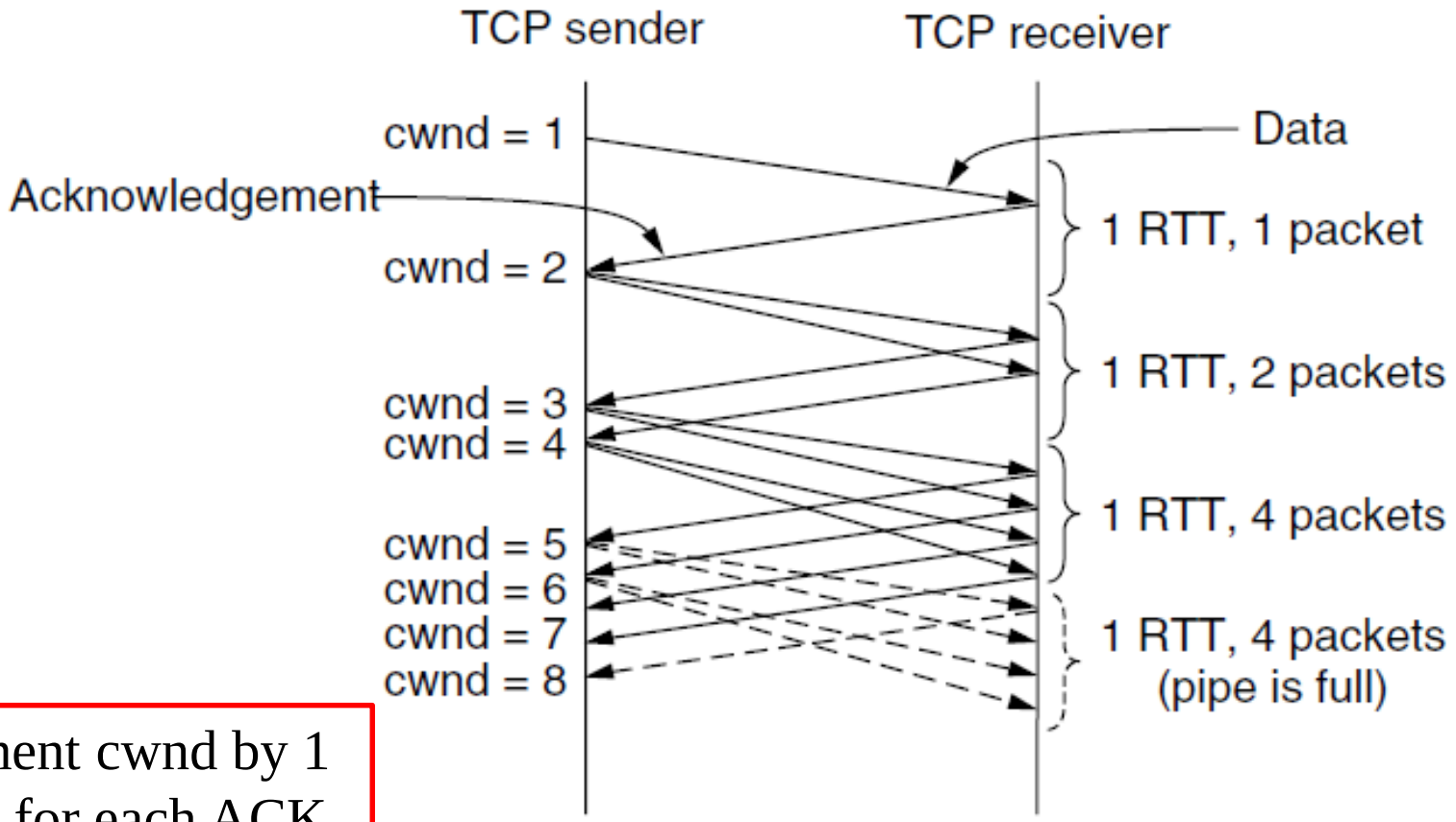


Figure 6-43. A burst of packets from a sender and the returning ack clock.

Slow-Start (Doubling) Timeline

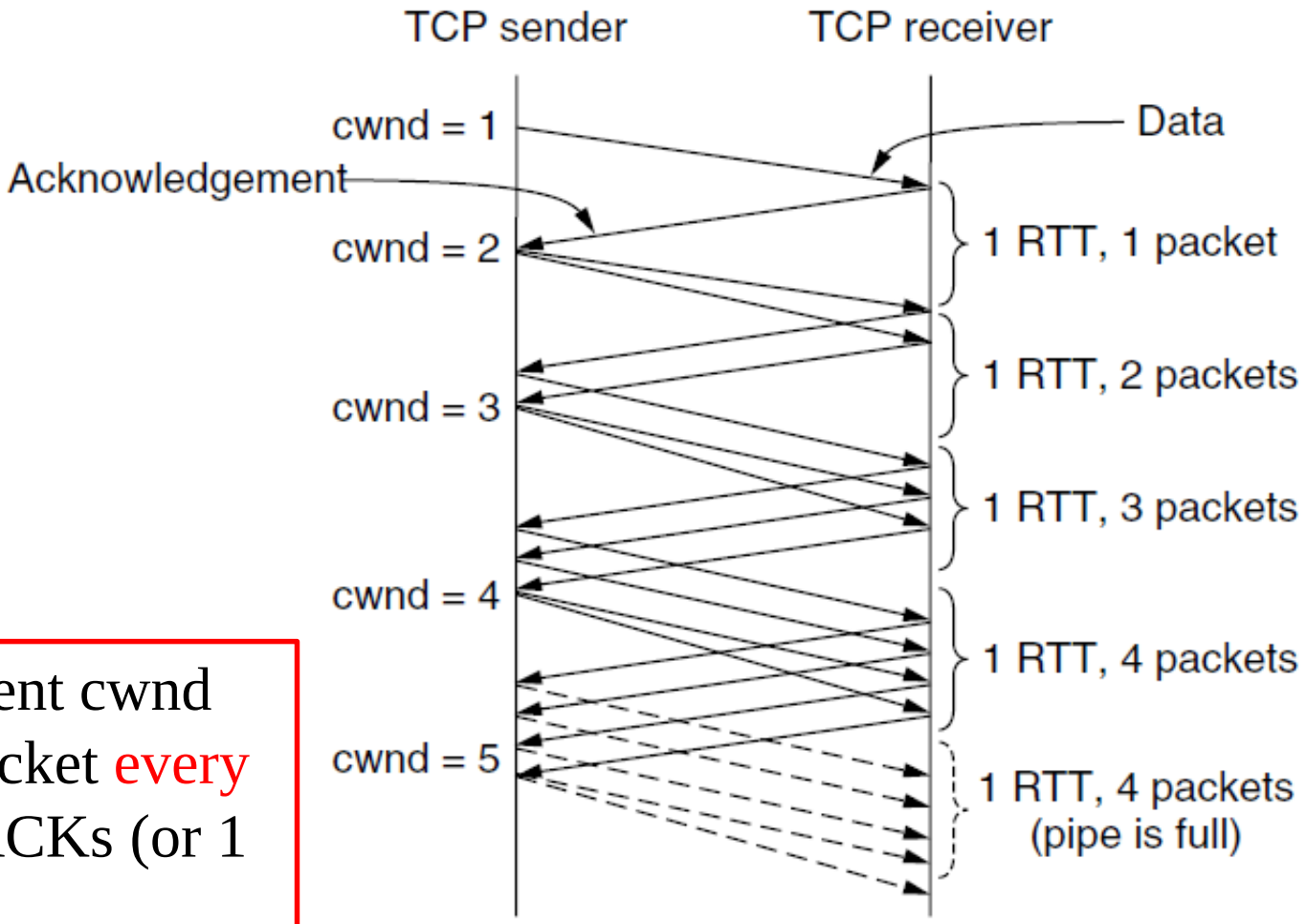


Increment cwnd by 1 packet for each ACK.

Figure 6-44. Slow start from an initial congestion window of one segment.

每收到一个ACK，就增加一个数据包，也就是一个变两个。

Additive Increase Timeline



Increment cwnd by 1 packet every cwnd ACKs (or 1 RTT)

Figure 6-45. Additive increase from an initial congestion window of one segment.

从上图看完成一个完整的RTT，才增加一个数据包。如cwnd = 3时，只有收到三个ACKs，才增加一个数据包。

TCP Congestion Control: Tahoe

- **TCP Tahoe** (1988)
  - The slow start threshold is 32 KB
  - The congestion window to 1KB for transmission 0. The window is increased every time a new acknowledgement. The congestion window grows exponentially until it hits the threshold (32KB).
  - After the threshold is passed, the window grows linearly. It is increased by one segment every RTT.

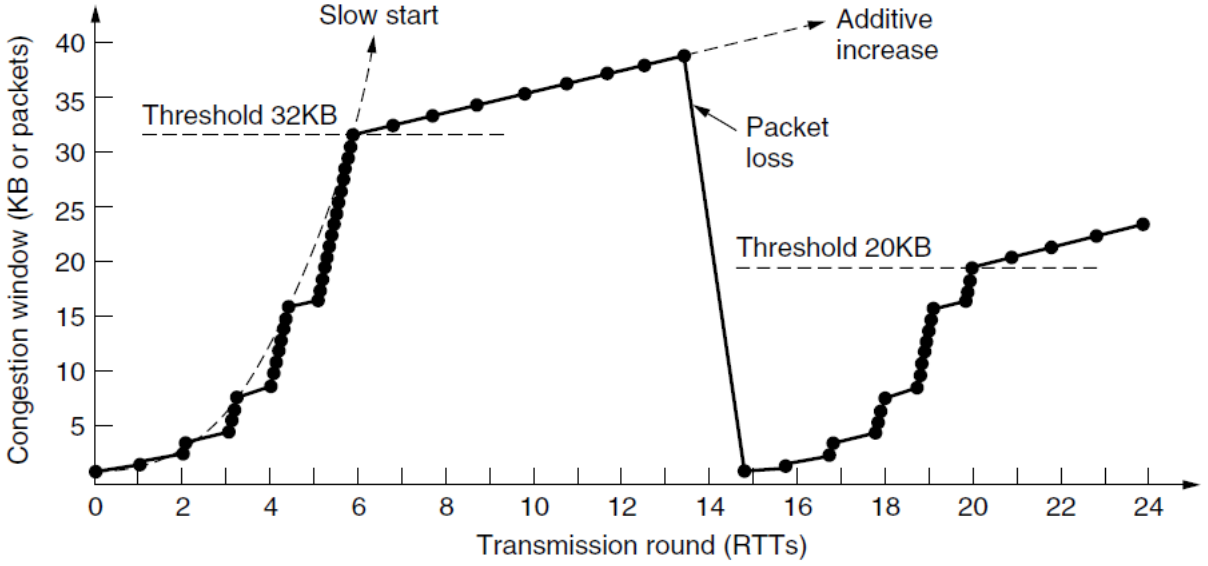
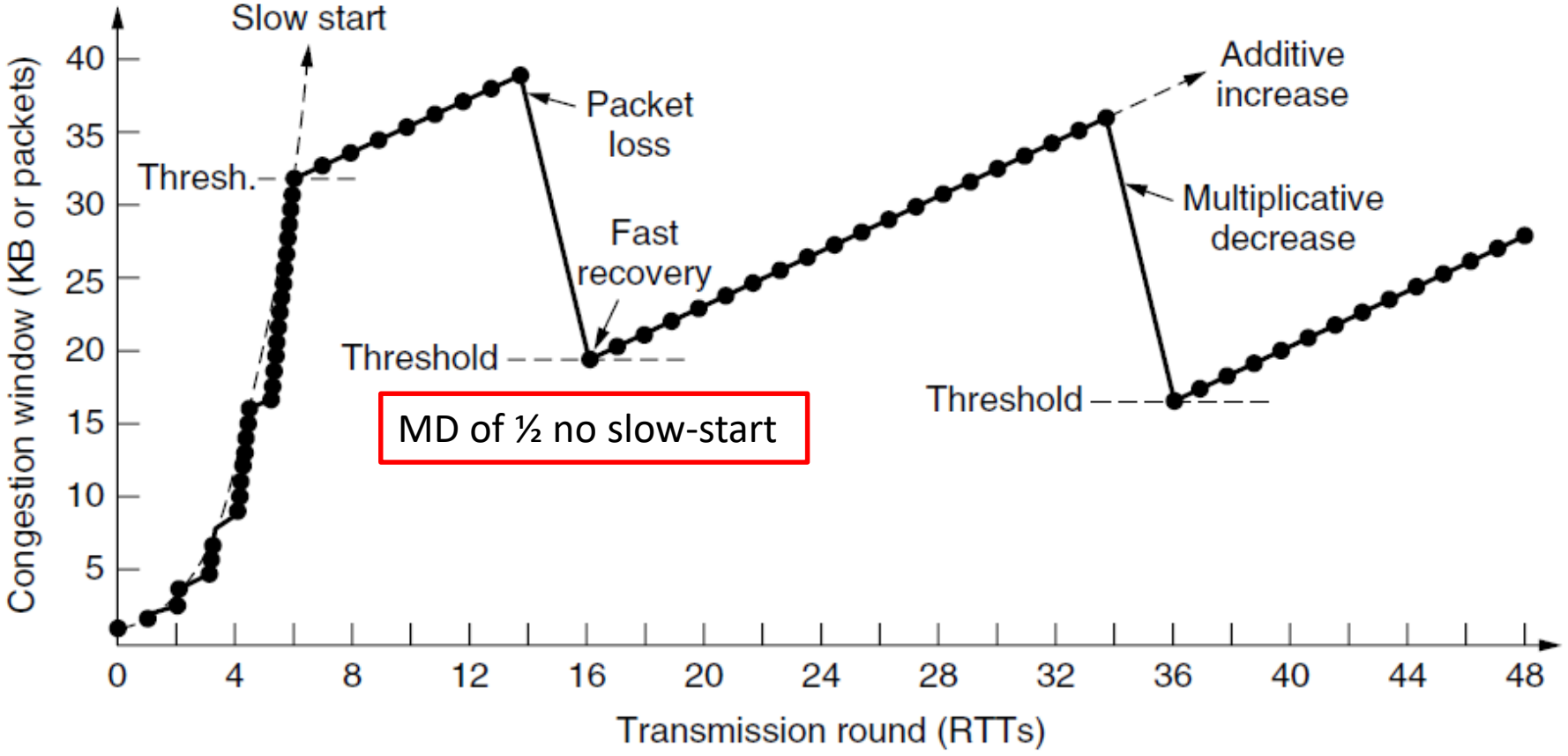


Figure 6-46. Slow start followed by additive increase in TCP Tahoe.

At time 13, one of packets is lost, then set the ssthresh = current cwnd (40)/2 = 20, and re-start slow start with cwnd = 1.

TCP Congestion Control: Reno

- TCP Reno with its mechanisms for adjusting the congestion control has formed the basis for TCP congestion control for more than two decades.



TCP Congestion Control: SACK

- Two larger changes have also affect TCP implementations
  - 1) **SACK** (**Selective ACKnowledgements**) lists up to three ranges of bytes that have been received. With this information, the sender can more directly decide which packets to retransmit and track the packets in flight to implement the congestion window.
  - With SACK, TCP can recover more easily from situations in which *multiple packets are lost at roughly the same time*, since the TCP sender knows which packets have not been received.
  - [RFC2883 and RFC3517]

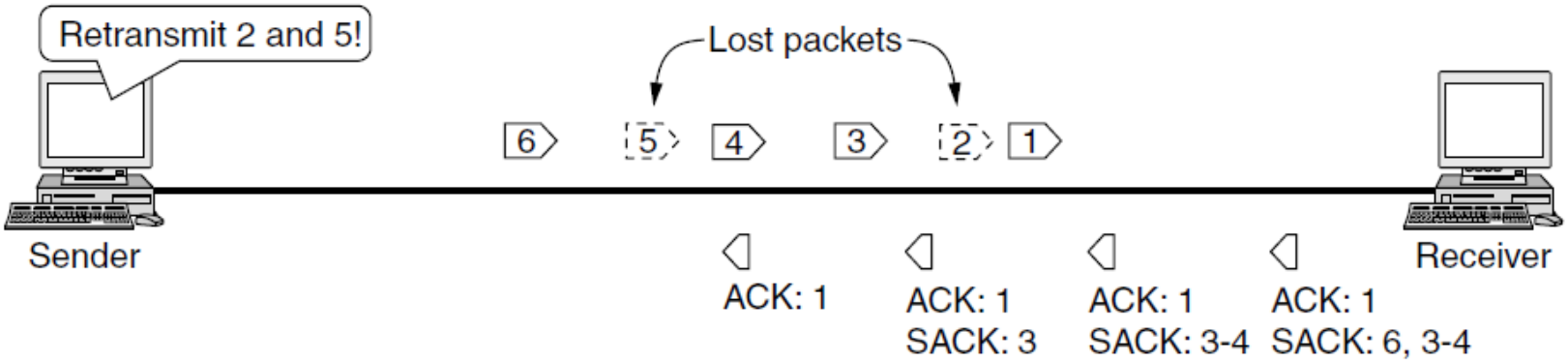


Figure 6-48. Selective acknowledgements.

APPLICATION LAYER

Important Network Applications

- DNS
- FTP
- Email
- Web

Important Network Applications

- DNS
  - To map a name to an IP address
  - UDP, port 53
  - The DNS database
- FTP
- Email
- Web

Domain Resource Records

- A resource record is a five-tuple. The format is as follows:

Domain_name Time_to_live Class Type Value

- The Domain_name tells the domain to which this record applies. Normally, many records exist for each domain and each copy of the database holds information about multiple domains. This field is thus **the primary search key** used to satisfy queries.
- The Time_to_live field gives an indication of how **stable** the record is. Information that is highly stable is assigned a large value; information that is highly volatile is assigned a small value.

Domain Resource Records

- The Class field. For Internet information, it is always IN. For non-Internet information, other codes can be used, but in practice these are rarely seen.
- The Type field

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

Figure 7-3. The principal DNS resource record types.

DNS Resolution

- DNS protocol lets a host resolve any host name (domain) to IP address
- If unknown, can start with the root nameserver and work down zones.

Example: flits.cs.vu.nl resolves robot.cs.washington.edu

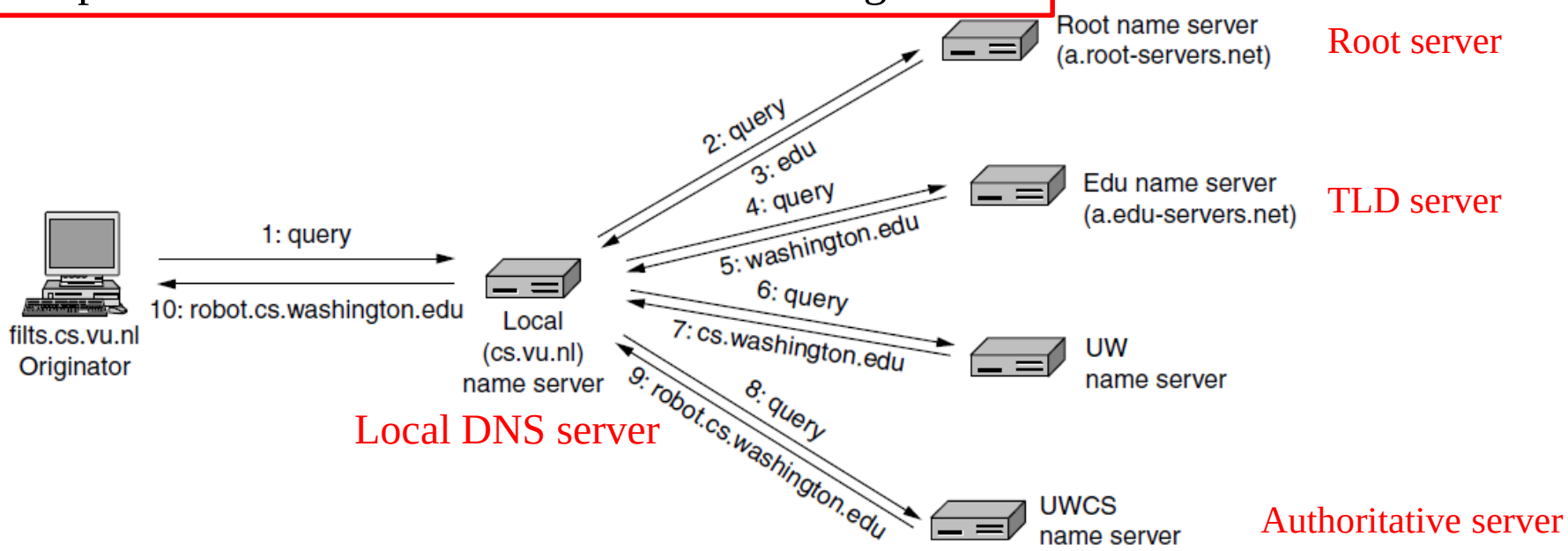


Figure 7-6. Example of a resolver looking up a remote name in 10 steps.

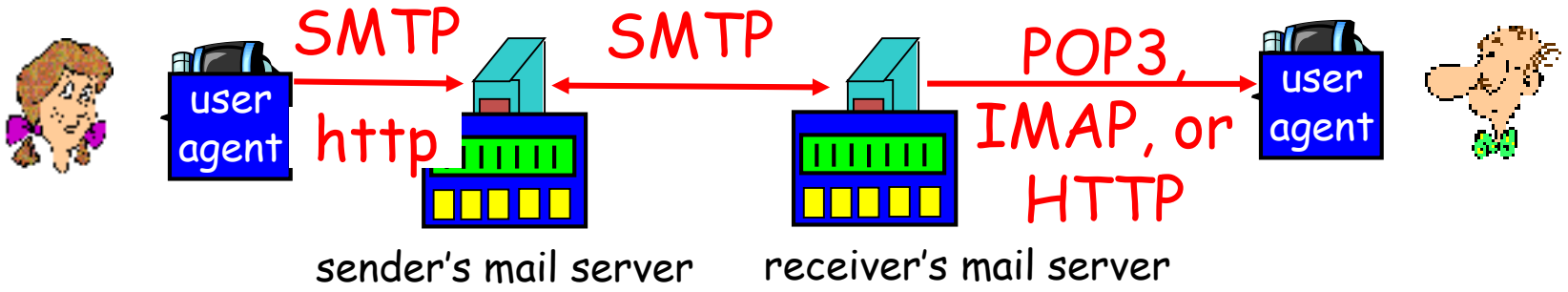
Iterative vs. Recursive Queries

- Recursive query
  - Nameserver completes resolution and returns the final answer
  - Lets server offload client burden (simple resolver) for manageability
  - Lets server cache over a pool of clients for better performance
    - E.g., flits → local nameserver (客户端-本地DNS服务器)
- Iterative query
  - Nameserver returns the answer or who to contact next for the answer
  - Lets server “file and forget”
  - Easy to build high load servers
    - local nameserver → all others (本地DNS服务器-外网)

Electronic Mail

- Email is an asynchronous communication medium.
- Three major components
  - User agents
  - Message transfer agents (mail servers)
  - Simple mail transfer protocol: **SMTP**
- It uses the reliable data transfer service of **TCP** to transfer mail from the sender's mail server to the recipient's mail server.
  - Port number: **25**
- It restricts the body (not just the headers) of all mail messages to simple **7-bit ASCII**.
- To obtain the messages is a **pull** operation, whereas **SMTP is a push protocol**.

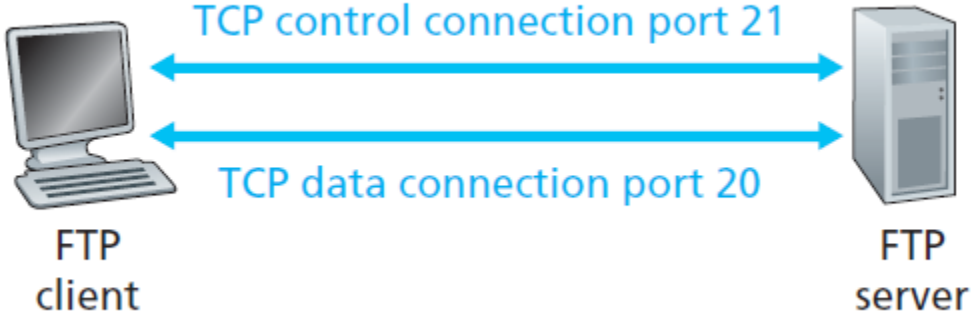
Mail Access Protocol – Final Delivery



- **SMTP: delivery/storage to receiver's server**
- **Mail access protocol: retrieval from server**
  - **POP: Post Office Protocol [RFC 1939] (port 110)**
    - **authorization (agent ↔ server) and download**
    - **Does not maintain state across POP sessions**
    - **Cannot manipulate emails at the server side**
  - **IMAP: Internet Mail Access Protocol [RFC 3501] (port 143)**
    - **more features (more complex)**
    - **manipulation of stored messages on server**
    - **Maintain state for the user**
  - **HTTP: Hotmail , Yahoo! Mail, etc.**
    - **Slow**

File Transfer: FTP [5] (III)

- FTP uses **two parallel TCP connections** to transfer a file, a control connection and a data connection.
  - The control connection is used for sending control information between the two hosts — information such as user identification, password, commands to change remote directory, and commands to "put" and "get" files.
    - FTP is said to send its control information **out-of-band**. (Because of this control connection (separate) FTP is "out-of-band" .)
    - The commands, from client to server, and replies, from server to client, are sent across the control connection in 7-bit ASCII format.
  - The data connection is used to actually send a file.



Important Network Applications

- DNS
- FTP
- Email
- Web
  - http protocol (stateless protocol)
    - Cookie
  - TCP, port 80; https, port 443
  - Web caching
  - html (css, cascading style sheet)

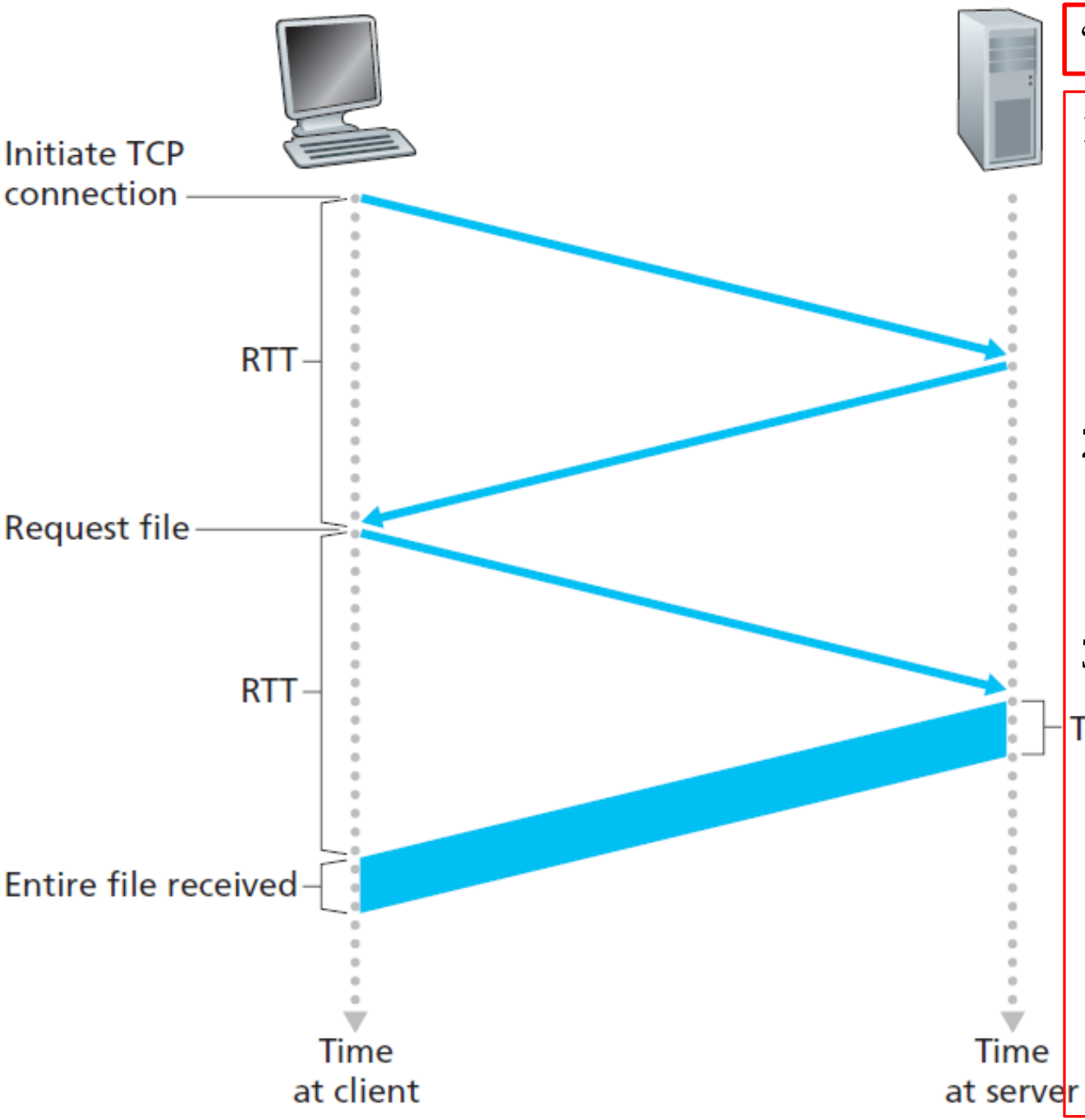
Fetching a Web page with HTTP (the Client Side)

- Start with the page URL

http://www.zju.edu.cn/index.html

The protocol	The server name	Page on the server

- Steps:
  - Resolve the server to IP address (DNS)
  - Set up TCP connection to the server
  - Send HTTP request for the page
  - (Await HTTP response for the page)
  - Execute / fetch embedded resources /render (不只是展示网页中内容，可能还要运行程序等。)
  - Clean up an idle TCP connections



"Three-way Handshake"

1) The browser (client) initiates a TCP connection by sending a small TCP segment to the server.
2) The server acknowledges and responds with a small TCP segment.
3) client sends the HTTP request message combined with the third part of the three-way handshake (the acknowledgement into the TCP connection).

Cookies (I)

- HTTP is a **stateless** protocol. [5]
- The simple request/response is not adequate when there are interactions between the users and Web sites. It is often desirable for a Web site to identify users.
  - Registration
  - E-commerce
- Solutions:
  - 1) IP address (but sometimes it does not work because of NAT, DHCP)
  - 2) Cookies (first implemented in the Netscape browser 1994, RFC2109, RFC2965)

Cookies (II)

- An HTTP cookie (web cookie, browser cookie) is a small piece of data (at most 4KB) that a server sends to the user's web browser
  - Typically, it is used to tell if two requests came from the same browser — keeping a user logged-in
  - It remembers stateful information for the stateless HTTP protocol.

- Cookies are mainly used for three purposes:
  - Session management
    - Logins, shopping carts, game scores, or anything else the server should remember.
  - Personalization
    - User preferences, themes, and other settings
  - Tracking
    - Recording and analyzing user behavior (DoubleClick, Google Analytics专门从事Web tracking生意的)

Cookies (III)

- A cookie may contain up to five fields
  - The **Domain** tells where the cookie came from
  - The **Path** is a path in the server's directory structure that identifies which parts of the server's file tree may use the cookie.
  - The **Content** field is where the cookie's content is stored.
  - The **Expires** field specifies when the cookie expires.
    - If this field is absent, the browser discards the cookie when it exits. (**nonpersistent cookie**)
    - If a time and date are supplied, the cookie is said to be a **persistent cookie**.
    - To remove a cookie from a client's hard disk, a server just sends it again, but with an expiration time in the past.
  - The **Secure** field can be sent to indicate that the browser may only return the cookie to a server using a secure transport. This feature is used for e-commerce.

Persistent Connections

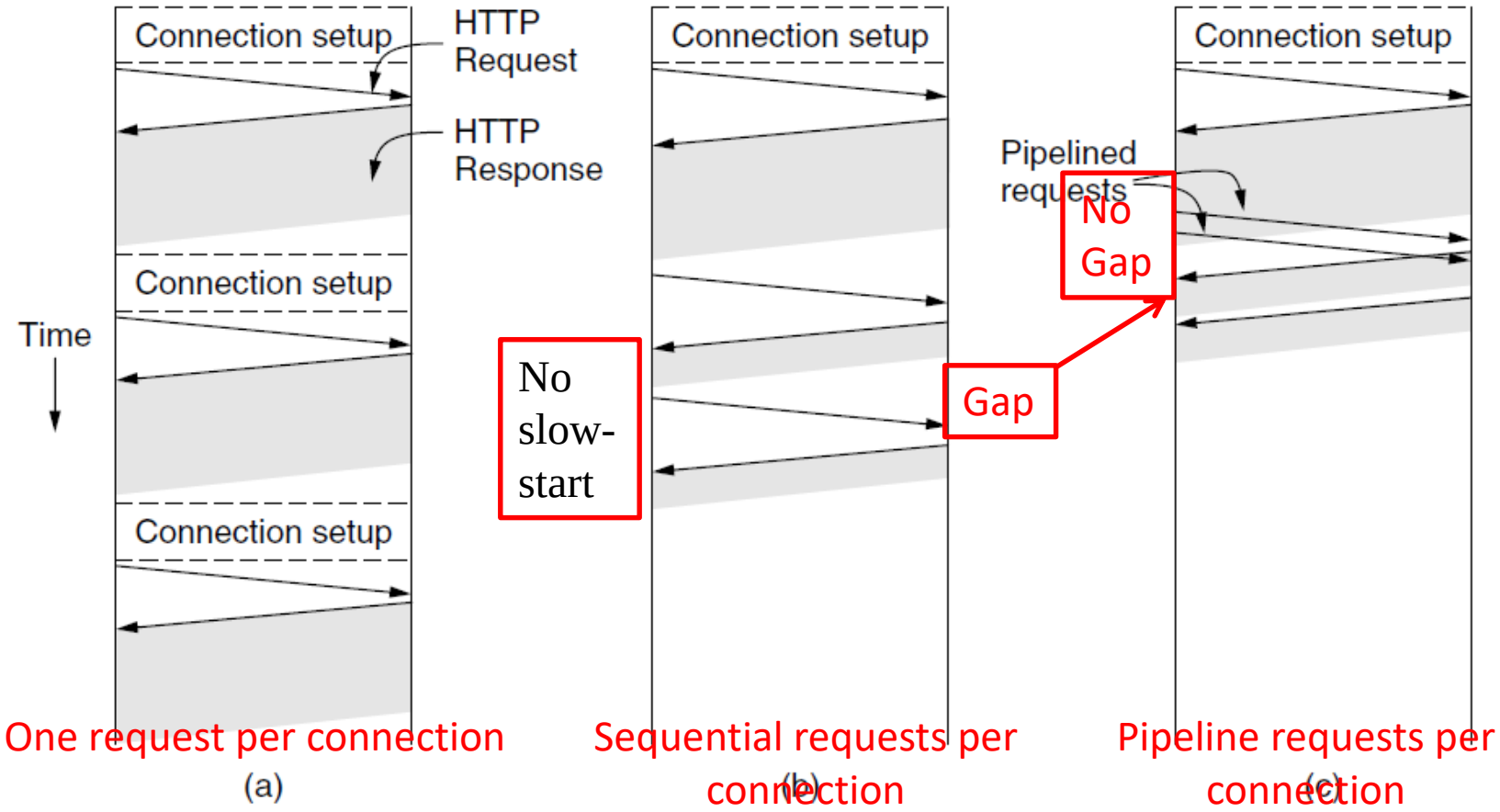


Figure 7-36. HTTP with (a) multiple connections and sequential requests. (b) A persistent connection and sequential requests. (c) A persistent connection and pipelined requests.

HTTP/1.1	HTTP/2
每个请求都需要单独建立一个 TCP 连接（虽然有 Keep-Alive 头字段可以在一定程度上保持连接复用，但效果有限）。	HTTP/2 采用了多路复用（Multiplexing）技术，它允许在一个 TCP 连接上同时发送多个请求和接收多个响应，而不需要像 HTTP/1.1 那样为每个请求单独建立连接。
HTTP/1.1 的请求和响应头部信息通常是未经压缩的文本格式，每次请求和响应都要完整地发送这些头部信息。	HTTP/2 采用了 HPACK 头部压缩算法，对请求和响应的头部信息进行高效压缩。它可以根据之前传输过的头部信息以及一些预设的规则，对重复出现的部分进行压缩处理，大大减少了头部信息占用的网络带宽。
HTTP/1.1 传输的数据格式是基于文本的，采用 ASCII 码进行编码。	HTTP/2 引入了二进制分帧层（Binary Framing Layer），它将所有传输的数据（包括请求、响应以及它们的头部和主体部分）都转换为二进制格式进行传输。
HTTP/1.1 没有明确的请求优先级机制。当浏览器同时发送多个请求（比如加载一个网页时，同时请求图像、脚本、样式表等资源），服务器会按照接收到请求的先后顺序来处理，无法根据资源的重要性或紧急程度进行有针对性的处理。	HTTP/2 具备明确的请求优先级机制。客户端（如浏览器）可以在发送请求时为不同的请求设置优先级，服务器收到这些请求后，会根据设置的优先级来安排处理顺序，优先处理重要性高、紧急程度高的请求，从而更合理地分配资源，提高用户体验。
HTTP/1.1 没有完善的流控制机制。	HTTP/2 建立了完善的流控制机制，通过窗口大小调整等方式来控制数据传输的速度。在网络拥塞时，它可以根据实际情况适当缩小窗口大小，减少数据传输量，避免过度占用网络资源；在网络状况良好时，又可以适当扩大窗口大小，加快数据传输速度，保证了网络传输的稳定性和高效性。

HTTP Message Format: Request

- Originally a simple protocol, with many options added over time
  - Text-based (ASCII) commands: request lines, header lines
  - The request line has three fields: the **method** field, the **URL** field, and the **HTTP version** field.
- Methods used in the **request**

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

Figure 7-37. The built-in HTTP request methods.

- The **GET** method requests the server to send the page.
- The **POST** method is used when a user fills out *a form*. It uploads the data to the server. The server then does something with the data that depends on the URL.
- The **PUT** method allows a user to upload an object to a specific path (directory) on a specific Web server.

HTTP Request Message Example

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/4.0
Accept-language: fr
```

1) **The request line** has three fields: *the method field*, *the URL field*, *and the HTTP version field*.

GET /somedir/page.html HTTP/1.1

2) The subsequent lines are called **header lines**

♦ Host: www.someschool.edu (specifies the host on which the object resides)

♦ Connection: close (the browser is telling the server that it doesn't want to bother with persistent connections; it wants the server to close the connection after sending the request object.)

♦ User-agent: Mozilla/4.0 (specifies the user agent, that is, the browser type that is making the request to the server.)

HTTP Message Format: Response

- Each request gets a **response** consisting of **a status line**, and possibly additional information.
- The status line contains a three-digit status code telling whether the request was satisfied and, if not, why not. The 1st digit is used to divide the responses into **five** major groups

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

Figure 7-38. The status code response groups.

HTTP Response Message Example

```
HTTP/1.1 200 OK
Connection: close
Date: Sat, 07 Jul 2007 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Sun, 6 May 2007 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data ...)
```

The response message has *three sections*: An initial **status line**, six **header lines** and then **the entire body**.

♦The status line has three fields: the protocol version, a status code, and a corresponding status message. (HTTP/1.1 200 OK)

♦The header lines (Connection, Date, Server, Last-Modified, Content-Length, Content-Type)

For example, the server uses the "Connection: Close" header line to tell the client that it is going to close the TCP connection after sending the message.

♦ The entire body: data

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
...	Set ... in boldface
...	Set ... in italics
...	Center ... on the page horizontally
...	Brackets an unordered (bulleted) list
...	Brackets a numbered list
	Starts a list item (there is no)
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
<img src="...">	Displays an image here
 ... 	Defines a hyperlink

Dynamic Web Pages

- Dynamic web page is the result of program execution
  - E-commerce, library catalogs, stock market, reading and sending email.
    - For example, a map service that lets user to enter a street address and presents a corresponding map of the location.

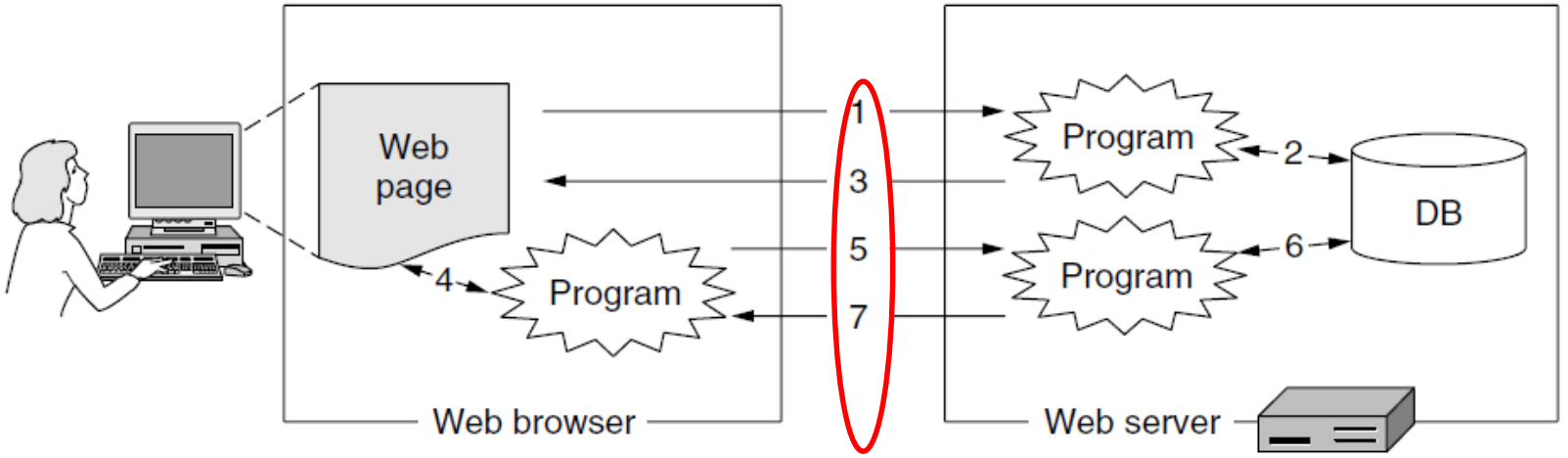


Figure 7-29. Dynamic pages.

1. request; 2. consults a database to generate the appropriate page; 3. return it to the browser; 4. update the page (zoom in or out) need more data; 5. request to the server; 6. retrieve more information; 7. return a response.

Server-Side Dynamic Web Page Generation

- Several APIs (Application Programming Interface) for handling dynamic page requests
  - **CGI** (Common Gateway Interface) provides an interface to allow Web servers to talk to back-end programs and scripts that can accept input (e.g., from forms) and generate HTML pages in response. — CGI scripts
    - RFC 3875
    - These programs usually be written in a script language, Python, Ruby, Perl.
  - To embed little scripts inside HTML pages and have them be executed by the server itself to generate the page. — embedded **PHP**
    - PHP (In PHP, after the user clicked on the submit button, the browser collects the information into a long string and sends it off to the server as a request for a PHP page.)
  - **JSP** (JavaServer Pages) is similar to PHP but written in Java programming language.

Client-Side Dynamic Web Page Generation

- Neither PHP nor CGI can respond to mouse movements or interact with users directly. For this purpose, it is necessary to have scripts embedded in HTML pages that are executed on the client machine rather than the server machine.
  - Starting with HTML 4.0, such scripts are permitted using **the tag <script>** — **dynamic HTML** (example Fig. 7-31)
- The most popular scripting language for the client side is **JavaScript**.
  - JavaScript has almost nothing to do with the Java programming language.
- VBScript
- Applets (These are small Java programs that have been compiled into machine instructions for a virtual computer called the JVM (Java Virtual Machine))

NETWORK SECURITY

Main Points (I)

- Kerckhoff's principle: All algorithms must be public; only the keys are secret.
  - The longer the key, the higher the work factor the cryptanalyst has to deal with.
- Encryption methods can be divided into *two categories*: **substitution ciphers** and **transposition ciphers**.
- In **symmetric-key** algorithms, they use **the same key** for encryption and decryption.
  - DES
  - Triple DES
  - AES (Rijndael)
  - RC4
  - RC5

Main Points (II)

- Public-key cryptography requires each user to have two keys: **a public key**, used by the entire world for encrypting message to be sent to that user, and **a private key**, which the user needs for decrypting messages.
  - RSA
- Digital signatures
  - Symmetric-key signatures
  - Public-key signatures
  - Message digests
    - SHA-1 or MD5
  - The birthday attack

Main Points (III)

- Applications
  - Communication security
    - IPsec
    - Firewalls: They are network layer devices, but they peek at the transport and application layers to do their filtering.
    - VPN
    - Wireless Security

Main Points (IV)

- Authentication protocols
  - **reflection attack**
  - 1) The Diffie-Hellman Key Exchange (shared key)
    - **The man-in-the-middle attack**
    - **Replay attack**
      - The solution to replay attack: time stamp and nonce
      - The Needham-Schroeder authentication protocol (1978)
      - The Otway-Rees protocol
  - 2) Kerberos: all clocks are fairly well synchronized, three servers (authentication server, ticket grant server, server)
  - 3) Public-key

Main Points (IV)

- Email security (**PGP**)
- Web Security
  - DNSSEC
  - **SSL/TLS** (HTTPs, port: 443)

References

- [1] A.S. Tanenbaum, and D.J. Wetherall, Computer Networks, 6th Edition, Prentice Hall, 2011.

- [2] J. F. Kurose and K.W. Ross, Computer Networking — A Top-down Approach, 5th Edition, Pearson Education Inc., 2010.