IKT204 || All Review Questions

# Chapter 1 - Review Questions and Problems

Exercise 1.1  (R11)Suppose there is exactly one packet switch between a sending host and a receiving host. The transmission rates between the sending host and the switch and between the switch and the receiving host are ***R*1** and ***R*2**, respectively. Assuming the switch uses store-and-forward packet switching, what is the total end-to-end delay to send a packet of length *L*? (Ignore queuing, propagation delay, and processing delay.

Exercise 1.2 (R23)What are the five layers in the Internet protocol stack? What are the principal responsibilities of each of these layers?

Exercise 1.3  (R24)What do encapsulation and de-encapsulation mean? Why are they needed in a layered protocol stack?

Exercise 1.4  (R25)Which layers in the Internet protocol stack does a router process? Which layers does a link-layer switch process? Which layers does a host process?

Exercise 1.5  (P6)Propagation delay and transmission delay are two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate ***R***bps. Suppose that the two hosts are separated by ***m***meters, and suppose the propagation speed along the link is ***s***meters/sec. Host A is to send a packet of size ***L***bits to Host B.

**a)** Express the propagation delay, ***d*prop**, in terms of ***m***and ***s***.  
**b)** Determine the transmission time of the packet, ***d*trans**, in terms of ***L***and ***R***.  
**c)** Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.  
**d)** Suppose Host A begins to transmit the packet at time***t*= 0**. At time ***t*= *d*trans**, where is the last bit of the packet?  
**e)** Suppose ***d*prop** is greater than ***d*trans**. At time ***t*= *d*trans**, where is the first bit of the packet?  
**f)** Suppose ***d*prop** is less than ***d*trans**. At time ***t*= *d*trans**, where is the first bit of the packet?  
**g)** Suppose ***s***= 2.6 **.** 108 m/s, ***L***= 1250 bytes, and ***R***= 200 Mbps. Find the distance ***m***so that ***d*prop** equals ***d*trans**.

Exercise 1.6  (P10)Consider the network illustrated in Figure 1.16. Assume the two hosts on the left of the figure start transmitting packets of 1500 bytes at the same time towards Router B. Suppose the link rates between the hosts and the Router A is 10 Mbps. One link has a 2.0 milliseconds propagation delay and the other has a 2.1 milliseconds propagation delay. Link rate between Router A and Router B is 50 Mbps. Will queuing delay occur in Router A? Support your answer with calculations.

Exercise 1.7  (P24)Consider a user who needs to upload 125 MB of data to a server. The user has a 100 Mbps Wi-Fi connection. How long is the upload time?

# Chapter 2 - Review Questions and Problems

## Exercise 2.1 (R3)

For a communication session between a pair of processes, which process is the client and which is the server?

## Exercise 2.2 (R5)

What information is used by a process running on one host to identify a process running on another host?

## Exercise 2.3 (R12)

How can websites keep track of users? Do they always need to use cookies?

## Exercise 2.4 (R16)

Suppose Alice, with a Web-based e-mail account (such as Hotmail or Gmail), sends a message to Bob, who accesses his mail from his mail server (using POP or IMAP). Discuss how the message gets from Alice’s host to Bob’s host. Be sure to list the series of application-layer protocols that are used to move the message between the two hosts.

## Exercise 2.5 (R18)

What is the HOL blocking (“først-i-køen”-blokkering) issue in HTTP/1.1? How does HTTP/2 attempt to solve it?

## Exercise 2.6 (P4)

Consider the following string of ASCII characters that were sent when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

GET /cs453/index.html HTTP/1.1Host: gaia.cs.umass.eduUser-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax)Accept:ext/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/plain;q=0.8,image/png,\*/\*;q=0.5  
Accept-Language: en-us,en;q=0.5Accept-Encoding: zip,deflateAccept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7Keep-Alive: 300Connection:keep-alive  
*<cr><lf>       (empty line)*

1. What is the URL of the document requested by the browser?
2. What version of HTTP is the browser running?
3. Does the browser request a non-persistent or a persistent connection?
4. What is the IP address of the host on which the browser is running?
5. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

## Exercise 2.7 (P18)

1. What is a *whois* database?
2. Use a *whois* databases on the Internet to obtain the names of two DNS servers (*for eksempel navneserverne til* uia.no). Indicate which *whois* databases you used.
3. Use *nslookup* on your local host to send DNS queries to three DNS servers: your local DNS server and the two DNS servers you found in part (b). Try querying for Type A, NS, and MX reports. Summarize your findings.
4. Use *nslookup* to find a Web server that has multiple IP addresses. Does the Web server of your institution (school or company) have multiple IP addresses?

## Exercise 2.8 (P22)

**Ch. 2.6 (30 points)**. Consider distributing a file of ***F*** = 500 MB to N peers, where M is SI standard (base 10) 106 and not binary (base 2) 220 = 10242. The server has an upload rate of ***us*** = 1000 Mbps, and each peer has a download rate of ***di*** = 80 Mbps (i.e., ***dmin***= ***di***) and an upload rate of ***ui***. For ***N*** = 15, 300, and 6 000 and ***ui*** = 2 Mbps, 10 Mbps, and 50 Mbps, prepare a table giving the minimum distribution time for each of the combinations of ***N*** and ***ui*** for both client-server distribution and P2P distribution.

# Chapter 3A - Review Questions and Problems

## Exercise 3A.1 (R3)

How is a UDP socket fully identified?

## Exercise 3A.2 (R4)

Describe why an application developer might choose to run an application over UDP rather than TCP.

## Exercise 3A.3 (R6)

Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

## Exercise 3A.4 (R7)

Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B each send a UDP segment to Host C with destination port number 6789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

## Exercise 3A.5 (P1)

Suppose Client A requests a web page from Server S through HTTP (which uses TCP) and its socket is associated with port 33000.

1. What are the source and destination ports for the segments sent from A to S?
2. What are the source and destination ports for the segments sent from S to A?
3. Can Client A contact to Server S using UDP as the transport protocol?

# Chapter 3B - Review Questions and Problems

## Exercise 3B.1

How many sockets does a UDP server need to serve N clients?

## Exercise 3B.2 (R3)

How is a UDP socket fully identified? What about a TCP socket? What is the difference between the full identification of both sockets?

## Exercise 3B.3 (R26)

If a TCP server were to support N simultaneous connections, each from a different client, how many sockets would the TCP server need?

## Exercise 3B.4 (R10)

What is the purpose of timers in TCP?

## Exercise 3B.5 (R15)

Suppose Host A sends two TCP segments back-to-back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.

1. How much data is in the first segment?
2. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

## Exercise 3B.6 (P15)

Consider the cross-country example shown in Figure 3.17. Suppose that the round-trip time RTT is 33 milliseconds, the channel transmission rate ***R*** is 1.2 Gbps, and the size of a packet is 1500 bytes, including both header fields and data. ACK packets are small, and their transmission delay can be ignored.

1. How big is channel utilization (in percent) when a **stop-and-wait protocol** is used?
2. How many unacknowledged segments in transit are permissible to achieve a channel-utilization greater than 98 percent when a **pipelined protocol** is used?
3. What is this in terms of window size, i.e., the range of permissible sequence numbers for transmitted but not yet acknowledged packets?

## Exercise 3B.7 (P27)

Host A and B are communicating over a TCP connection. Host B has already received from Host A all bytes up through byte 96. Suppose Host A then sends two segments to Host B. The first and the second segments contain 40 and 80 bytes of data, respectively. In the first segment, the sequence number is 97, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

1. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?
2. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?
3. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?
4. Suppose the two segments sent by A arrive in order at B. The first acknowledgment arrives after the first timeout interval. What is the sequence number of the next segment that A will transmit?

# UDP socket-programmering

## UDP Task 1: server

Make a program **in C** that will wait for a request from a client, and send a response back to the same client. The server shall make a simple conversion:

* Server receives a string, and converts it to uppercase, using for example toupper().
* Server sends the modified string back to the client.

The server must receive a request and send a response using a socket assigned a specific port number, for instance 50000.

You can build on this:

#include <stdio.h>

#include <winsock2.h>

#include <ws2tcpip.h> // InetPton()

#define BUF\_SIZE 50

void printIPaddr(struct sockaddr\_in addr) {

printf("%d.%d.%d.%d",

addr.sin\_addr.S\_un.S\_un\_b.s\_b1, addr.sin\_addr.S\_un.S\_un\_b.s\_b2,

addr.sin\_addr.S\_un.S\_un\_b.s\_b3, addr.sin\_addr.S\_un.S\_un\_b.s\_b4);

}

int UDP\_server(const short nPortNumber) {

WSADATA wsa;

SOCKET socketServer = INVALID\_SOCKET;

struct sockaddr\_in addrServer;

struct sockaddr\_in addressReceived;

int nFromlen = sizeof(addressReceived);

int nRes = 0;

char sReceivedString[BUF\_SIZE] = { 0 };

if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0) {

printf("Failed. Error Code : %d", WSAGetLastError());

return 1;

}

printf("WinSock initialised.\n");

socketServer = socket(AF\_INET, SOCK\_DGRAM, 0); // Initialize IPv4/UDP socket instance

if(socketServer == INVALID\_SOCKET) {

printf("Could not create socket : %d", WSAGetLastError());

return 1;

}

printf("Socket created.\n");

addrServer.sin\_family = AF\_INET; // IPv4

addrServer.sin\_addr.s\_addr = INADDR\_ANY;

addrServer.sin\_port = htons(nPortNumber); // little-endian -> big-endian, host-to-network short

nRes = bind(socketServer, (struct sockaddr\*)&addrServer, sizeof(addrServer));

if (nRes != 0) {

printf("Bind failed with error code : %d", WSAGetLastError());

closesocket(socketServer);

return 1;

}

printf("Listening to incoming UDP packets at port number %d ...\n", ntohs(addrServer.sin\_port)); // big-endian -> little-endian, network-to-host short

nRes = recvfrom(socketServer, sReceivedString, BUF\_SIZE, 0, (struct sockaddr \*) &addressReceived, &nFromlen);

if (nRes == SOCKET\_ERROR) {

printf("recvfrom failed with error code: %d\n", WSAGetLastError());

closesocket(socketServer);

return 1;

}

printf("Packet received ... \n");

printf(" Source IP address: ");

printIPaddr(addressReceived);

printf("\n");

printf(" Source port number: %d\n", ntohs(addressReceived.sin\_port)); // Big-endian -> little-endian, network-to-host short

printf(" Received string: %s. Received %d bytes\n", sReceivedString, strlen(sReceivedString));

printf("\n");

// Insert code to send response...

closesocket(socketServer);

WSACleanup();

return 0;

}

UDP.2. client  
Make a client program **in C** that does the following:

* The client reads a line of characters from its keyboard, and sends this string to server.
* The client then waits for a response from the server. When receiving the response, the client prints the string on the screen.

The destination source port number in the request from the client must match the port number of the receiving UDP socket at the server.

You can build on this:

#include <stdio.h>

#include <winsock2.h>

#include <ws2tcpip.h> // InetPton()

#define BUF\_SIZE 50

void printIPaddr(struct sockaddr\_in addr) {

printf("%d.%d.%d.%d",

addr.sin\_addr.S\_un.S\_un\_b.s\_b1, addr.sin\_addr.S\_un.S\_un\_b.s\_b2,

addr.sin\_addr.S\_un.S\_un\_b.s\_b3, addr.sin\_addr.S\_un.S\_un\_b.s\_b4);

}

int UDP\_client(const short nPortNumber, const char sTargetAddress[]) {

WSADATA wsa;

SOCKET socketClient;

struct sockaddr\_in addressServer;

struct sockaddr\_in addressReceived;

char sSendMessage[BUF\_SIZE] = {0};

char sReceivedMessage[BUF\_SIZE] = {0};

int nNum = 0;

// WSAStartup to initiate WS2\_32.dll, and initialise the WSADATA structures .

if (WSAStartup(MAKEWORD(2,2),&wsa) != 0) {

printf("WSAStartup failed: %d\n", WSAGetLastError());

return 1;

}

printf("Winsock initialised.\n");

socketClient = socket(AF\_INET, SOCK\_DGRAM, 0 ); // Initialize IPv4/UDP socket instance

if(socketClient == INVALID\_SOCKET) {

printf("Could not create socket: %d\n" , WSAGetLastError());

closesocket(socketClient);

WSACleanup();

return 1;

}

printf("Socket created.\n");

InetPton(AF\_INET, \_\_TEXT(sTargetAddress), & addressServer.sin\_addr); // convert IP address as char array into binary

addressServer.sin\_family = AF\_INET; // IPv4

addressServer.sin\_port = htons(nPortNumber); // little-endian -> big-endian, host-to-network short

if(sendto(socketClient, sSendMessage, strlen(sSendMessage), 0, (struct sockaddr\*)&addressServer, sizeof(addressServer) ) < 0) {

printf("sendto failed: %d\n", WSAGetLastError());

closesocket(socketClient);

WSACleanup();

return 1;

}

// sendto generates a random source port number because it is not specified

struct sockaddr\_in sin1;

socklen\_t len1 = sizeof(sin1);

getsockname(socketClient, (struct sockaddr \*)&sin1, &len1); // retrieves the local IP address+port number

printf("\nRequest sent ...\n");

printf(" Source port number: %d\n", ntohs(sin1.sin\_port)); // big-endian -> little-endian, network-to-host short

printf(" Target IP address: "); printIPaddr(addressServer); printf("\n");

printf(" Target port number: %d\n", ntohs(addressServer.sin\_port)); // big-endian -> little-endian, network-to-host short

printf(" Sent string: %s\n", sSendMessage);

printf("\n");

// Insert code to receive response from server....

// Show response data

printf("Response received ...\n");

printf(" Source IP address: "); printIPaddr(addressReceived); printf("\n");

printf(" Source port number: %d\n", ntohs(addressReceived.sin\_port)); // big-endian -> little-endian, network-to-host short

printf(" Received string: %s. Received %d bytes\n", sReceivedMessage, nNum);

printf("\n");

closesocket(socketClient);

WSACleanup();

return 0;

}

# Chapter 4 - Review Questions and Problems

## Exercise 4.1 (R3)

We made a distinction between the forwarding function and the routing function performed in the network layer. What are the key differences between routing and forwarding?

## Exercise 4.2 (R17)

Suppose Host A sends Host B a TCP segment encapsulated in an IP datagram. When Host B receives the datagram, how does the network layer in Host B know it should pass the segment (that is, the payload of the datagram) to TCP rather than to UDP or to some other upper-layer protocol?

## Exercise 4.3 (R18)

What field in the IP header can be used to ensure that a packet is forwarded through no more than N routers?

## Exercise 4.4 (R21)

How many IP addresses does a router have?

## Exercise 4.5 (R25)

Suppose an application generates 40 bytes of data that gets encapsulated in a TCP segment and then an IP datagram. (Assume that the IP “options” field is 0 bytes.)

1. a) What percentage of the datagram is overhead and what percentage is application data?
2. b) What is the overhead percentage if UDP is used instead of TCP?

## Exercise 4.6 (P17)

Suppose datagrams are limited to 1,500 bytes (including header) between source Host A and destination Host B. Assuming a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain how you computed your answer.

## Exercise 4.7 (R31)

How makes “tunneling” it possible to send IPv6 packets over IPv4 routers?

## Exercise 4.8 (P8, P13)

Consider an IP network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

|  |  |
| --- | --- |
| **Destination Address Range** | **Link Interface** |
| 11010000  00000000  00000000  00000000 through 11010000  00111111  11111111  11111111 | 0 |
| 11010000  01000000  00000000  00000000 through 11010000  01000001  11111111  11111111 | 1 |
| 11010000  01000010  00000000  00000000 through 11010001  11011111  11111111  11111111 | 2 |
| otherwise | 3 |

1. Provide a routing table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.   
   (Hint. The third entry needs to be split to cover the given address range.)
2. Rewrite this routing table using the CIDR a.b.c.d/x notation instead of the binary string notation.
3. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

|  |
| --- |
| 11010001  11110111  01110011  00111001 |
| 11010000  00001100  10101101  11100100 |
| 11010000  01000000  11000011  01110010 |
| 11010001  10001111  01100111  00011111 |

## Exercise 4.9 (P14)

Consider a subnet with prefix 128.119.40.128/26.

1. a) How many IP addresses does this prefix represent?
2. b) What are the starting and ending IP addresses of the address block represented by this prefix?

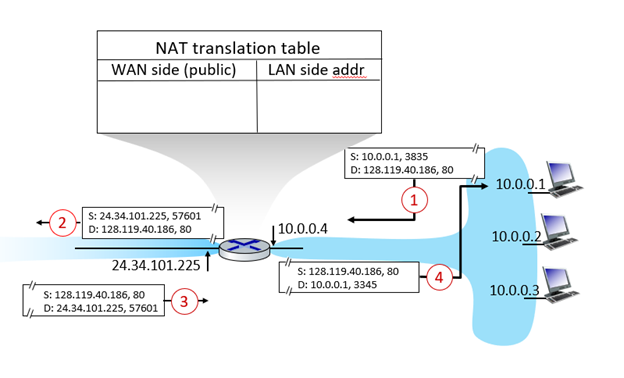
## Exercise 4.10 (P14)

Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

## Exercise 4.11 (P18 variant)

Consider the packet flow with NAT in the figure below

1. Provide the corresponding entry in the NAT translation table.
2. Suppose that the ISP instead assigns the router the address 143.43.131.225 and that the subnet address of the home network is 192.168.0/24. Assign addresses to all interfaces in the home network.



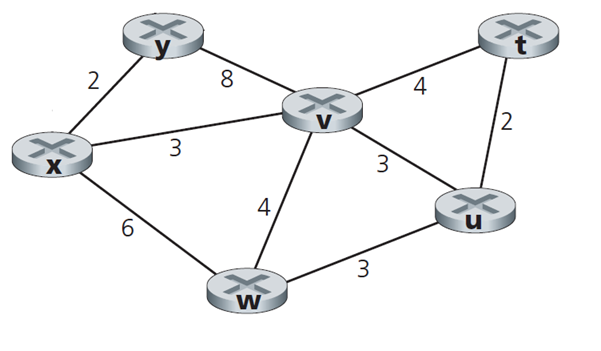
# Chapter 5 - Review Questions and Problems

## Exercise 5.1 (R3)

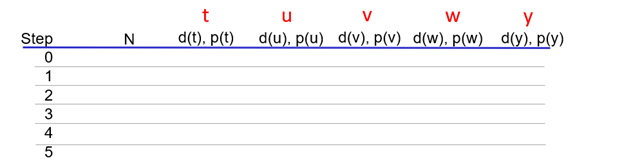
Compare and contrast the properties of a centralized (global) and a distributed *routing* ***algorithm***. Give an example of a *routing* ***protocol*** that takes a centralized and a decentralized approach.

## Exercise 5.2 (P1, P2 variant)

Consider the following network.



1. Use Dijkstra’s shortest-path algorithm to compute the shortest path from **node** **x** to all network nodes. Show how the algorithm works by computing a table below.
2. Use Dijkstra’s shortest-path algorithm to compute the shortest path from **node** **t** to all network nodes.



## Exercise 5.3 (R7)

Why are different inter-AS and intra-AS protocols used in the Internet?

## Exercise 5.4 (R8)

When an OSPF router sends its link state information, it is sent only to those routers that are directly attached neighbors or to all routers in the same autonomous system? Justify your answer.

## Exercise 5.5 (R10)

Define and contrast the following terms: subnet, prefix, and BGP route.

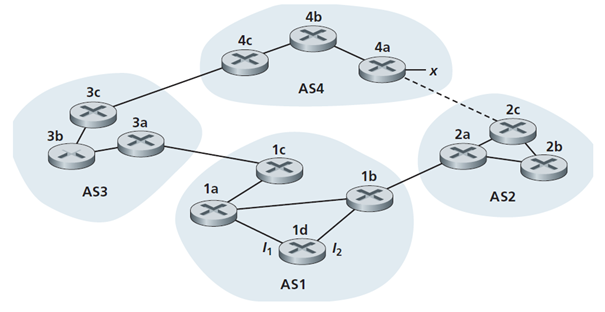
## Exercise 5.6 (R11)

How does BGP use the NEXT-HOP attribute? How does it use the AS-PATH attribute?

## Exercise 5.7 (P14)

Consider the network shown below. Suppose that AS1, AS2, AS3 and AS4 are running OSPF for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4.

1. Router **3c** learns about prefix ***x*** from which routing protocol: OSPF, eBGP, or iBGP?
2. Router **3a** learns about ***x*** from which routing protocol?
3. Router **1c** learns about ***x*** from which routing protocol?
4. Router **1d** learns about ***x*** from which routing protocol?
5. Once router **1d** learns about prefix ***x***, it will put an entry *(****x,I****)* in its forwarding table, where ***I*** is either interface ***I1*** or ***I2***. Will ***I*** be equal to ***I1*** or ***I2*** ? Explain why in one sentence.
6. Now suppose that a link between AS2 and AS4 (shown by the dotted line) is established, and that router **1d** learns that prefix ***x*** is accessible via AS2 as well as via AS3. Will ***I*** be set to ***I1*** or ***I2*** ? Explain why in one sentence.



# Chapter 6 - Review Questions and Problems

## Exercise 6.1 (R2)

If all the links on the Internet were to provide reliable delivery service, would the TCP reliable delivery service be redundant? Why or why not?

## Exercise 6.2 (R10)

Suppose nodes A, B, and C each attach to the same broadcast LAN (through their adapters).

1. If A sends thousands of IP datagrams to B with each encapsulating frame addressed to the MAC address of B, will C’s adapter process these frames? If so, will C’s adapter pass the IP datagrams in these frames to C’s network layer?
2. How would your answers change if A sends frames with the MAC broadcast address?

## Exercise 6.3

1. Why is an ARP query sent within a broadcast frame?
2. Why is an ARP response sent within a frame with a specific destination MAC address?

## Exercise 6.4 (R15 variant)

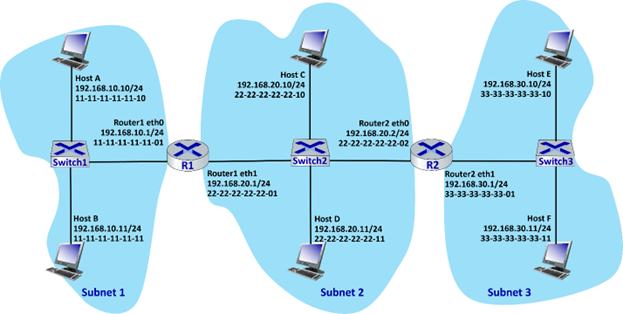
Each host and router have an ARP table in its memory (an ARP cache).

1. What is the content of this table?
2. What is the main reason for having such a table?

## Exercise 6.5 (P14 variant)

Consider three LANs interconnected by two routers, as shown in figure below.

1. Host F is sending an IP datagram to Host A. Suppose all ARP tables are up to date. Describe and enumerate all the steps, as done for the single-router example in Section 6.4.1.
2. Assume that the ARP table in the sending host F is empty (and the other tables are up to date). Describe and enumerate all the steps.
3. Subnet 2 is interconnected other networks via two routers. Can the hosts in this subnet have two default gateway routers? Discuss how traffic out of this subnet is handled. (Hint: Can DHCP provide some options?)



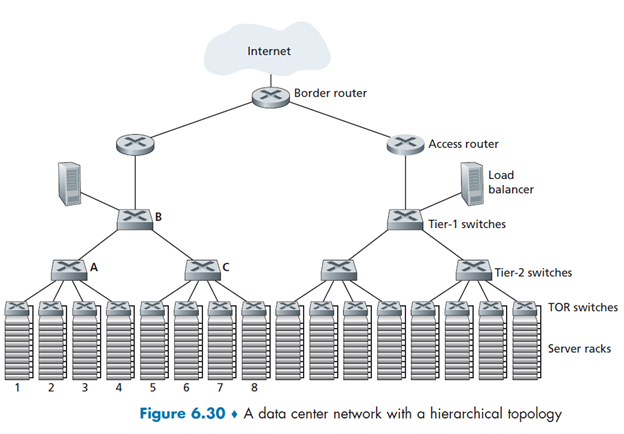
## Exercise 6.6 (P26)

Let us consider the operation of a learning switch in the context of a network in which 6 nodes labeled A through F are star connected into an Ethernet switch. Suppose that *(i)* B sends a frame to E, *(ii)* E replies with a frame to B, *(iii)* A sends a frame to B, *(iv)* B replies with a frame to A. The switch table is initially empty. Show the state of the switch table before and after each of these events. For each of these events, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers

## Exercise 6.7 (P33)

Consider the hierarchical network in Figure 6.30 below and suppose that the data center needs to support e-mail and video distribution among other applications. Suppose *four* racks of servers are reserved for e-mail and *four* racks are reserved for video. For each of the applications, all four racks must lie below a single tier-2 switch since the tier-2 to tier-1 links do not have sufficient bandwidth to support the intra-application traffic. For the e-mail application, suppose that for 99.9 percent of the time only three racks are used, and that the video application has identical usage patterns.

1. For what fraction of time does the e-mail application need to use a fourth rack? How about for the video application?
2. Assuming e-mail usage and video usage are independent, for what fraction of time do both applications need their fourth rack?



# Chapter 7 - Review Questions and Problems

## Exercise 7.1 (R1)

1. What does it mean for a wireless network to be operating in “infrastructure mode”?
2. If the network is **not** in infrastructure mode, what mode of operation is it in, and what is the difference between that mode of operation and infrastructure mode?

## Exercise 7.2 (R3)

What are the differences between the following types of wireless channel impairments: path loss (norsk: signaltap), multipath propagation, interference from other sources?

## Exercise 7.3 (R4)

As a mobile node gets farther and farther away from a base station, path loss (norsk: signaltap) occurs causing a reduced signal-to-noise ratio (SNR). What two actions could a base station take to ensure that the loss probability of a transmitted frame does not increase?

## Exercise 7.4 (R5)

Describe the role of the beacon frames in 802.11.

## Exercise 7.5 (R6)

An access point periodically sends beacon frames. What is the content of the beacon frames?

## Exercise 7.6 (R9)

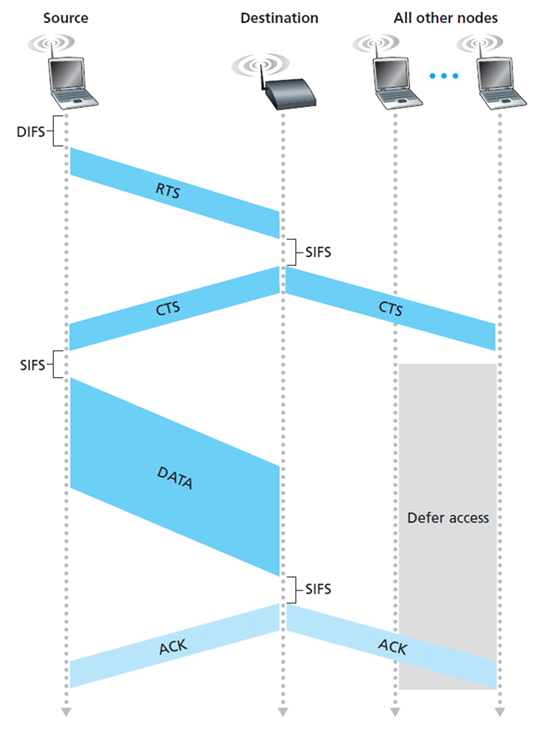
What are the two main purposes of a CTS frame?

## Exercise 7.7

After selecting an AP (by means of its SSID), a wireless host sends a request frame to the AP and the AP responds with a response frame. When connected to an AP, the host will want to join the subnet to which the AP belongs. What does the host do next?

## Exercise 7.8 (P7 variant)

Suppose an 802.11b wireless host is configured to reserve the channel with the RTS/CTS sequence. Suppose this host wants to transmit 1000 bytes of data, and all other stations are idle at this time.



We have the following assumptions

* The transmission rate in the wireless network is 11 Mbps.
* For 802.11b, DIFS (DCF Interframe Spacing) is 50 microseconds and SIFS (Short Interframe Spacing) is 10 microseconds.
* A 812.11 frame without data is 34 bytes long (see Fig. 7.13 in the textbook). The CTS, RTS and ACK control frames are frames without data.

What will the total time to transmit the frame and receive the acknowledgement as a function of SIFS and DIFS, ignoring propagation delay?

## Exercise 7.9 (P9)

Power is a precious resource in mobile devices, and thus the 802.11 standard provides power-management capabilities that allow 802.11 nodes to minimize the amount of time that their sense, transmit, and receive functions and other circuitry need to be “on.” In 802.11, a node is able to explicitly alternate between sleep and wake states. Explain in brief how a node communicates with the AP to perform power management.

# Chapter 8 - Review Questions and Problems

## Exercise 8.1

Explain Kerckhoff's principle with regard to cryptographic keys and encryption/decryption algorithms.

## Exercise 8.2

What security property is provided by symmetric key and asymmetric key encryption?

## Exercise 8.3

1. Consider symmetric key encryption. If an attacker gets hold of both a ciphertext and its underlying encrypted plaintext, can the attacker easily deduce the encryption key?
2. Consider asymmetric key encryption. If an attacker gets hold of both a ciphertext and its underlying encrypted plaintext, can the attacker easily deduce the private key?

## Exercise 8.4

All block ciphers and certain public key encryption algorithms are deterministic, meaning that the same plaintext and key produces the same ciphertext.

1. Are block ciphers resistant to chosen plaintext attacks?
2. Are deterministic public key encryption algorithms resistant to chosen plaintext attacks?

## Exercise 8.5

Checksums and CRC32 codes provide integrity protection concerning transmission errors.

1. Do hash functions provide integrity protection in this regard?
2. Do hash functions provide integrity protection regarding a (passive) eavesdropper?
3. Do hash functions provide integrity protection regarding an active adversary?

## Exercise 8.6

Consider a hash function whose output length is 160 bits. Why is it difficult to find two different input messages that produce the same hash, i.e. h(m1)=h(m2)?

## Exercise 8.7

What are the different purposes of link-layer MAC and cryptographic MAC?

## Exercise 8.8 (variant R15)

After being victim of a fraudulent individual, Alice has decided to ensure the integrity of data she sends to her clients.

1. Would integrity protection make the clients confident that Alice is the originator and sender of her data?
2. Regarding protecting against fraudulent individuals, what is suitable of using 1) message digests, 2) MACs, or 3) digital signatures? Explain what is achieved by each alternative.

## Exercise 8.9 (variant R6)

Suppose that a team of 10 persons are using symmetric key encryption to communicate securely with the other members of the team, without anybody else in the team being able to decrypt.

1. How many keys need to be shared in total among the team members?
2. Given the pairwise shared keys, how can four people in the group communicate securely together using symmetric key encryption (without anybody else in the team being able to decrypt).
3. Suppose that the group uses public key encryption instead. What is the answer to a) and b) in this case?

## Exercise 8.10

What are the advantages of using public key encryption compared to secret key encryption? What potential security problem does public keys introduce, and how can this be mitigated?

## Exercise 8.11

Why do digital signatures have a stronger security property (non-repudiation) compared to MACs (message authentication)?

## Exercise 8.12

A digital certificate contains the name of the holder, a public key, and a digital signature.

1. Whose signature is it?
2. What is the purpose of digital signature certificates?
3. Does a single digital certificate guarantee this security goal?

## Exercise 8.13

Replay attacks are a significant category of attacks.

1. What security property prevents replay attacks?
2. How can replay attacks be prevented?

## Exercise 8.14

What is the difference between key transfer and key agreement? Why is key agreement preferable?

## Exercise 8.15 (R21)

1. What is the purpose of the random nonces in the TLS handshake?
2. How do they achieve this purpose?

# Additional problems from the textbook

## Chapter 1 Review Questions

*SECTION 1.1*

R1. What is the difference between a host and an end system? List several different types of end systems. Is a Web server an end system?

R2. The word *protocol* is often used to describe diplomatic relations. How does Wikipedia describe diplomatic protocol?

R3. Why are standards important for protocols?

*SECTION 1.2*

R4. List six access technologies. Classify each one as home access, enterprise access, or widearea wireless access.

R5. Is HFC transmission rate dedicated or shared among users? Are collisions possible in a downstream HFC channel? Why or why not?

R6. List the available residential access technologies in your city. For each type of access, provide the advertised downstream rate, upstream rate, and monthly price.

R7. What is the transmission rate of Ethernet LANs?

R8. What are some of the physical media that Ethernet can run over?

R9. Dial-up modems, HFC, DSL and FTTH are all used for residential access. For each of these access technologies, provide a range of ­transmission rates and comment on whether the transmission rate is shared or dedicated.

R10. Describe the most popular wireless Internet access technologies today. ­Compare and contrast them.

*SECTION 1.3*

R11. Suppose there is exactly one packet switch between a sending host and a receiving host. The transmission rates between the sending host and the switch and between the switch and the receiving host are *R*1 and *R*2, respectively. Assuming that the switch uses store-and-forward packet switching, what is the total end-to-end delay to send a packet of length *L*? (Ignore queuing, propagation delay, and processing delay.)

R12. What advantage does a circuit-switched network have over a packet-switched network?

What advantages does TDM have over FDM in a circuit-switched network?

R13. Suppose users share a 2 Mbps link. Also suppose each user transmits continuously at 1 Mbps when transmitting, but each user transmits only 20 percent of the time. (See the

discussion of statistical multiplexing in **Section 1.3** .)

1. When circuit switching is used, how many users can be supported?
2. For the remainder of this problem, suppose packet switching is used. Why will there be essentially no queuing delay before the link if two or fewer users transmit at the same time? Why will there be a queuing delay if three users transmit at the same time?
3. Find the probability that a given user is transmitting.
4. Suppose now there are three users. Find the probability that at any given time, all three users are transmitting simultaneously. Find the fraction of time during which the queue grows.

R14. Why will two ISPs at the same level of the hierarchy often peer with each other? How does an IXP earn money?

R15. Some content providers have created their own networks. Describe Google’s network.

What motivates content providers to create these networks?

*SECTION 1.4*

R16. Consider sending a packet from a source host to a destination host over a fixed route. List the delay components in the end-to-end delay. Which of these delays are constant and which are variable?

R17. Visit the Transmission Versus Propagation Delay applet at the companion Web site. Among the rates, propagation delay, and packet sizes available, find a combination for which the sender finishes transmitting before the first bit of the packet reaches the receiver. Find another combination for which the first bit of the packet reaches the receiver before the sender finishes transmitting.

R18. How long does it take a packet of length 1,000 bytes to propagate over a link of distance 2,500 km, propagation speed 2.5⋅108 m/s, and transmission rate 2 Mbps? More generally, how long does it take a packet of length *L* to propagate over a link of distance *d*, propagation speed *s*, and transmission rate *R* bps? Does this delay depend on packet length? Does this delay depend on transmission rate?

R19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1=500 kbps, R2=2 Mbps, and R3=1 Mbps.

1. Assuming no other traffic in the network, what is the throughput for the file transfer?
2. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?
3. Repeat (a) and (b), but now with *R*2 reduced to 100 kbps.

R20. Suppose end system A wants to send a large file to end system B. At a very high level, describe how end system A creates packets from the file. When one of these packets arrives to a router, what information in the packet does the router use to determine the link onto which the packet is forwarded? Why is packet switching in the Internet analogous to driving from one city to another and asking directions along the way?

R21. Visit the Queuing and Loss applet at the companion Web site. What is the maximum emission rate and the minimum transmission rate? With those rates, what is the traffic intensity? Run the applet with these rates and determine how long it takes for packet loss to occur. Then repeat the experiment a second time and determine again how long it takes for packet loss to occur. Are the values different? Why or why not?

*SECTION 1.5*

R22. List five tasks that a layer can perform. Is it possible that one (or more) of these tasks could be performed by two (or more) layers?

R23. What are the five layers in the Internet protocol stack? What are the principal responsibilities of each of these layers?

R24. What is an application-layer message? A transport-layer segment? A network-layer datagram? A link-layer frame?

R25. Which layers in the Internet protocol stack does a router process? Which layers does a link-layer switch process? Which layers does a host process?

*SECTION 1.6*

R26. What is the difference between a virus and a worm?

R27. Describe how a botnet can be created and how it can be used for a DDoS attack.

R28. Suppose Alice and Bob are sending packets to each other over a computer network.

Suppose Trudy positions herself in the network so that she can capture all the packets sent by Alice and send whatever she wants to Bob; she can also capture all the packets sent by Bob and send whatever she wants to Alice. List some of the malicious things Trudy can do from this position.

Problems

P1. Design and describe an application-level protocol to be used between an automatic teller machine and a bank’s centralized computer. Your protocol should allow a user’s card and password to be verified, the account balance (which is maintained at the centralized computer) to be queried, and an account withdrawal to be made (that is, money disbursed to the user).

Your protocol entities should be able to handle the all-too-common case in which there is not enough money in the account to cover the withdrawal. Specify your protocol by listing the messages exchanged and the action taken by the automatic teller machine or the bank’s centralized computer on transmission and receipt of messages. Sketch the operation of your protocol for the case of a simple withdrawal with no errors, using a diagram similar to that in

**Figure 1.2** . Explicitly state the assumptions made by your protocol about the underlying end-toend transport service.

P2. **Equation 1.1** gives a formula for the end-to-end delay of sending one packet of length *L* over *N* links of transmission rate *R*. Generalize this formula for sending *P* such packets back-toback over the *N* links.

P3. Consider an application that transmits data at a steady rate (for example, the sender generates an *N*-bit unit of data every *k* time units, where *k* is small and fixed). Also, when such an application starts, it will continue running for a relatively long period of time. Answer the following questions, briefly justifying your answer:

1. Would a packet-switched network or a circuit-switched network be more appropriate for this application? Why?
2. Suppose that a packet-switched network is used and the only traffic in this network comes from such applications as described above. Furthermore, assume that the sum of the application data rates is less than the capacities of each and every link. Is some form of congestion control needed? Why?

P4. Consider the circuit-switched network in **Figure 1.13** . Recall that there are 4 circuits on each link. Label the four switches A, B, C, and D, going in the clockwise direction.

1. What is the maximum number of simultaneous connections that can be in progress at any one time in this network?
2. Suppose that all connections are between switches A and C. What is the maximum number of simultaneous connections that can be in progress?
3. Suppose we want to make four connections between switches A and C, and another four connections between switches B and D. Can we route these calls through the four links to accommodate all eight ­connections?

P5. Review the car-caravan analogy in **Section 1.4** . Assume a propagation speed of 100 km/hour.

1. Suppose the caravan travels 150 km, beginning in front of one tollbooth, passing through a second tollbooth, and finishing just after a third tollbooth. What is the end-to-end delay?
2. Repeat (a), now assuming that there are eight cars in the caravan instead of ten.

P6. This elementary problem begins to explore propagation delay and transmission delay, two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate *R* bps. Suppose that the two hosts are separated by *m* meters, and suppose the propagation speed along the link is *s* meters/sec. Host A is to send a packet of size *L* bits to Host B.

1. Express the propagation delay, *d*prop, in terms of *m* and *s*.
2. Determine the transmission time of the packet, *d*trans, in terms of *L* and *R*.
3. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.
4. Suppose Host A begins to transmit the packet at time t=0. At time t= *d*trans, where is the last bit of the packet?
5. Suppose *d*prop is greater than *d*trans. At time t=dtrans, where is the first bit of the packet?
6. Suppose *d*prop is less than *d*trans. At time t=dtrans, where is the first bit of the packet?
7. Suppose s=2.5⋅108, L=120 bits, and R=56 kbps. Find the distance *m* so that *d*prop equals

*d*trans.

P7. In this problem, we consider sending real-time voice from Host A to Host B over a packetswitched network (VoIP). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 56-byte packets. There is one link between Hosts A and B; its transmission rate is 2 Mbps and its propagation delay is 10 msec. As soon as Host A gathers a packet, it sends it to Host B. As soon as Host B receives an entire packet, it converts the packet’s bits to an analog signal. How much time elapses from the time a bit is created (from the original analog signal at Host A) until the bit is decoded (as part of the analog signal at Host B)?

P8. Suppose users share a 3 Mbps link. Also suppose each user requires 150 kbps when transmitting, but each user transmits only 10 percent of the time. (See the discussion of packet

switching versus circuit switching in **Section 1.3** .)

1. When circuit switching is used, how many users can be supported?
2. For the remainder of this problem, suppose packet switching is used. Find the probability that a given user is transmitting.
3. Suppose there are 120 users. Find the probability that at any given time, exactly *n* users are transmitting simultaneously. (*Hint*: Use the binomial distribution.)
4. Find the probability that there are 21 or more users transmitting ­simultaneously.

P9. Consider the discussion in **Section 1.3** of packet switching versus circuit switching in which an example is provided with a 1 Mbps link. Users are generating data at a rate of 100 kbps when busy, but are busy generating data only with probability p=0.1. Suppose that the 1 Mbps link is replaced by a 1 Gbps link.

1. What is *N,* the maximum number of users that can be supported simultaneously under circuit switching?
2. Now consider packet switching and a user population of *M* users. Give a formula (in terms of *p*, *M*, *N*) for the probability that more than *N* users are sending data.

P10. Consider a packet of length *L* that begins at end system A and travels over three links to a destination end system. These three links are connected by two packet switches. Let *di*, *si*, and *Ri* denote the length, propagation speed, and the transmission rate of link *i,* for i=1,2,3. The packet switch delays each packet by *d*proc. Assuming no queuing delays, in terms of *di*, *si*, *Ri*,

(i=1,2,3), and *L,* what is the total end-to-end delay for the packet? Suppose now the packet is 1,500 bytes, the propagation speed on all three links is 2.5⋅108m/s, the transmission rates of all three links are 2 Mbps, the packet switch processing delay is 3 msec, the length of the first link is 5,000 km, the length of the second link is 4,000 km, and the length of the last link is 1,000 km.

For these values, what is the end-to-end delay?

P11. In the above problem, suppose R1=R2=R3=R and dproc=0. Further suppose the packet switch does not store-and-forward packets but instead immediately transmits each bit it receives before waiting for the entire packet to arrive. What is the end-to-end delay?

P12. A packet switch receives a packet and determines the outbound link to which the packet should be forwarded. When the packet arrives, one other packet is halfway done being transmitted on this outbound link and four other packets are waiting to be transmitted. Packets are transmitted in order of arrival. Suppose all packets are 1,500 bytes and the link rate is 2 Mbps. What is the queuing delay for the packet? More generally, what is the queuing delay when all packets have length *L*, the transmission rate is *R*, *x* bits of the currently-being-transmitted packet have been transmitted, and *n* packets are already in the queue?

P13.

1. Suppose *N* packets arrive simultaneously to a link at which no packets are currently being transmitted or queued. Each packet is of length *L* and the link has transmission rate *R*. What is the average queuing delay for the *N* packets?
2. Now suppose that *N* such packets arrive to the link every *LN/R* seconds. What is the average queuing delay of a packet?

P14. Consider the queuing delay in a router buffer. Let *I* denote traffic intensity; that is, I=La/R.

Suppose that the queuing delay takes the form IL/R(1−I) for I<1.

1. Provide a formula for the total delay, that is, the queuing delay plus the transmission delay.
2. Plot the total delay as a function of *L /R*.

P15. Let *a* denote the rate of packets arriving at a link in packets/sec, and let *µ* denote the link’s transmission rate in packets/sec. Based on the formula for the total delay (i.e., the queuing delay plus the transmission delay) derived in the previous problem, derive a formula for the total delay in terms of *a* and *µ*.

P16. Consider a router buffer preceding an outbound link. In this problem, you will use Little’s formula, a famous formula from queuing theory. Let *N* denote the average number of packets in the buffer plus the packet being transmitted. Let *a* denote the rate of packets arriving at the link.

Let *d* denote the average total delay (i.e., the queuing delay plus the transmission delay) experienced by a packet. Little’s formula is N=a⋅d. Suppose that on average, the buffer contains 10 packets, and the average packet queuing delay is 10 msec. The link’s transmission rate is 100 packets/sec. Using Little’s formula, what is the average packet arrival rate, assuming there is no packet loss?

P17.

1. Generalize **Equation 1.2** in **Section 1.4.3** for heterogeneous processing rates, transmission rates, and propagation delays.
2. Repeat (a), but now also suppose that there is an average queuing delay of *d*queue at each node.

P18. Perform a Traceroute between source and destination on the same continent at three different hours of the day.

|  |
| --- |
| A blue square with a white arrow  AI-generated content may be incorrect.  Using Traceroute to discover network paths and measure network delay |

1. Find the average and standard deviation of the round-trip delays at each of the three hours.
2. Find the number of routers in the path at each of the three hours. Did the paths change during any of the hours?
3. Try to identify the number of ISP networks that the Traceroute packets pass through from source to destination. Routers with similar names and/or similar IP addresses should be considered as part of the same ISP. In your experiments, do the largest delays occur at the peering interfaces between adjacent ISPs?
4. Repeat the above for a source and destination on different continents. Compare the intra-continent and inter-continent results.

P19.

1. Visit the site [**www.traceroute.org**](http://www.traceroute.org/) and perform traceroutes from two different cities in

France to the same destination host in the United States. How many links are the same

in the two traceroutes? Is the transatlantic link the same?

1. Repeat (a) but this time choose one city in France and another city in Germany.
2. Pick a city in the United States, and perform traceroutes to two hosts, each in a different city in China. How many links are common in the two traceroutes? Do the two traceroutes diverge before reaching China?

P20. Consider the throughput example corresponding to **Figure 1.20(b)** . Now suppose that there are *M* client-server pairs rather than 10. Denote *Rs*, *Rc*, and *R* for the rates of the server links, client links, and network link. Assume all other links have abundant capacity and that there is no other traffic in the network besides the traffic generated by the *M* client-server pairs. Derive a general expression for throughput in terms of *Rs*, *Rc*, *R*, and *M*.

P21. Consider **Figure 1.19(b)** . Now suppose that there are *M* paths between the server and the client. No two paths share any link. Path k(k=1,…,M) consists of *N* links with transmission rates R1k,R2k,…,RNk. If the server can only use one path to send data to the client, what is the maximum throughput that the server can achieve? If the server can use all *M* paths to send data, what is the maximum throughput that the server can achieve?

P22. Consider **Figure 1.19(b)** . Suppose that each link between the server and the client has a packet loss probability *p,* and the packet loss probabilities for these links are independent. What is the probability that a packet (sent by the server) is successfully received by the receiver? If a packet is lost in the path from the server to the client, then the server will re-transmit the packet. On average, how many times will the server re-transmit the packet in order for the client to successfully receive the packet?

P23. Consider **Figure 1.19(a)** . Assume that we know the bottleneck link along the path from the server to the client is the first link with rate *Rs* bits/sec. Suppose we send a pair of packets back to back from the server to the client, and there is no other traffic on this path. Assume each

packet of size *L* bits, and both links have the same propagation delay *d*prop.

1. What is the packet inter-arrival time at the destination? That is, how much time elapses from when the last bit of the first packet arrives until the last bit of the second packet arrives?
2. Now assume that the second link is the bottleneck link (i.e., Rc<Rs). Is it possible that the second packet queues at the input queue of the second link? Explain. Now suppose that the server sends the second packet *T* seconds after sending the first packet. How large must *T* be to ensure no queuing before the second link? Explain.

P24. Suppose you would like to urgently deliver 40 terabytes data from Boston to Los Angeles. You have available a 100 Mbps dedicated link for data transfer. Would you prefer to transmit the data via this link or instead use FedEx over-night delivery? Explain.

P25. Suppose two hosts, A and B, are separated by 20,000 kilometers and are connected by a direct link of R=2 Mbps. Suppose the propagation speed over the link is 2.5⋅108 meters/sec.

1. Calculate the bandwidth-delay product, R⋅dprop.
2. Consider sending a file of 800,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?
3. Provide an interpretation of the bandwidth-delay product.
4. What is the width (in meters) of a bit in the link? Is it longer than a ­football field?
5. Derive a general expression for the width of a bit in terms of the propagation speed *s,* the transmission rate *R,* and the length of the link *m*.

P26. Referring to problem P25, suppose we can modify *R*. For what value of *R* is the width of a bit as long as the length of the link?

P27. Consider problem P25 but now with a link of R=1 Gbps.

1. Calculate the bandwidth-delay product, R⋅dprop.
2. Consider sending a file of 800,000 bits from Host A to Host B. Suppose the file is sent continuously as one big message. What is the maximum number of bits that will be in the link at any given time?
3. What is the width (in meters) of a bit in the link?

P28. Refer again to problem P25.

1. How long does it take to send the file, assuming it is sent continuously?
2. Suppose now the file is broken up into 20 packets with each packet containing 40,000 bits. Suppose that each packet is acknowledged by the receiver and the transmission time of an acknowledgment packet is negligible. Finally, assume that the sender cannot send a packet until the preceding one is acknowledged. How long does it take to send the file?
3. Compare the results from (a) and (b).

P29. Suppose there is a 10 Mbps microwave link between a geostationary satellite and its base station on Earth. Every minute the satellite takes a digital photo and sends it to the base station.

Assume a propagation speed of 2.4⋅108 meters/sec.

1. What is the propagation delay of the link?
2. What is the bandwidth-delay product, R⋅dprop?
3. Let *x* denote the size of the photo. What is the minimum value of *x* for the microwave link to be continuously transmitting?

P30. Consider the airline travel analogy in our discussion of layering in **Section 1.5** , and the addition of headers to protocol data units as they flow down the protocol stack. Is there an equivalent notion of header information that is added to passengers and baggage as they move down the airline protocol stack?

P31. In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message* *segmentation*. **Figure 1.27** illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is 8⋅106 bits long that is to be sent from source to destination in **Figure 1.27** .

Suppose each link in the figure is 2 Mbps. Ignore propagation, queuing, and processing delays.

1. Consider sending the message from source to destination *without* message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?
2. Now suppose that the message is segmented into 800 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?
3. How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.

A close up of a text

AI-generated content may be incorrect.

A diagram of a glass container

AI-generated content may be incorrect.

*Figure 1.27 End-to-end message transport: (a) without message ­segmentation; (b) with message segmentation*

1. In addition to reducing delay, what are reasons to use message ­segmentation?
2. Discuss the drawbacks of message segmentation.

P32. Experiment with the Message Segmentation applet at the book’s Web site. Do the delays in the applet correspond to the delays in the previous problem? How do link propagation delays affect the overall end-to-end delay for packet switching (with message segmentation) and for message switching?

P33. Consider sending a large file of *F* bits from Host A to Host B. There are three links (and two switches) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of *S* bits each and adds 80 bits of header to each segment, forming packets of L=80 + *S* bits. Each link has a transmission rate of *R* bps. Find the value of *S* that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

P34. Skype offers a service that allows you to make a phone call from a PC to an ordinary phone. This means that the voice call must pass through both the Internet and through a telephone network. Discuss how this might be done.

## Chapter 2 Review Questions

*SECTION 2.1*

R1. List five nonproprietary Internet applications and the application-layer protocols that they use.

R2. What is the difference between network architecture and application architecture?

R3. For a communication session between a pair of processes, which process is the client and which is the server?

R4. For a P2P file-sharing application, do you agree with the statement, “There is no notion of client and server sides of a communication session”? Why or why not?

R5. What information is used by a process running on one host to identify a process running on another host?

R6. Suppose you wanted to do a transaction from a remote client to a server as fast as possible.

Would you use UDP or TCP? Why?

R7. Referring to **Figure 2.4** , we see that none of the applications listed in **Figure 2.4** requires both no data loss and timing. Can you conceive of an application that requires no data loss and that is also highly time-sensitive?

R8. List the four broad classes of services that a transport protocol can provide. For each of the service classes, indicate if either UDP or TCP (or both) provides such a service.

R9. Recall that TCP can be enhanced with SSL to provide process-to-process security services, including encryption. Does SSL operate at the transport layer or the application layer? If the application developer wants TCP to be enhanced with SSL, what does the developer have to do?

*SECTION 2.2–2.5*

R10. What is meant by a handshaking protocol?

R11. Why do HTTP, SMTP, and POP3 run on top of TCP rather than on UDP?

R12. Consider an e-commerce site that wants to keep a purchase record for each of its customers. Describe how this can be done with cookies.

R13. Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects?

Why?

R14. Telnet into a Web server and send a multiline request message. Include in the request message the *If-modified-since:* header line to force a response message with the *304 Not Modified* status code.

R15. List several popular messaging apps. Do they use the same protocols as SMS?

R16. Suppose Alice, with a Web-based e-mail account (such as Hotmail or Gmail), sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice’s host to Bob’s host. Be sure to list the series of application-layer protocols that are used to move the message between the two hosts.

R17. Print out the header of an e-mail message you have recently received. How many *Received:* header lines are there? Analyze each of the header lines in the message.

R18. From a user’s perspective, what is the difference between the download-and-delete mode and the download-and-keep mode in POP3?

R19. Is it possible for an organization’s Web server and mail server to have exactly the same alias for a hostname (for example, [***foo.com***](http://foo.com/))? What would be the type for the RR that contains the hostname of the mail server?

R20. Look over your received e-mails, and examine the header of a message sent from a user with a .edu e-mail address. Is it possible to determine from the header the IP address of the host from which the message was sent? Do the same for a message sent from a Gmail account.

*SECTION 2.5*

R21. In BitTorrent, suppose Alice provides chunks to Bob throughout a 30-second interval. Will Bob necessarily return the favor and provide chunks to Alice in this same interval? Why or why not?

R22. Consider a new peer Alice that joins BitTorrent without possessing any chunks. Without any chunks, she cannot become a top-four uploader for any of the other peers, since she has nothing to upload. How then will Alice get her first chunk?

R23. What is an overlay network? Does it include routers? What are the edges in the overlay network?

*SECTION 2.6*

R24. CDNs typically adopt one of two different server placement philosophies. Name and briefly describe them.

R25. Besides network-related considerations such as delay, loss, and bandwidth performance, there are other important factors that go into designing a CDN server selection strategy. What are they?

*SECTION 2.7*

R26. In Section 2.7, the UDP server described needed only one socket, whereas the TCP server needed two sockets. Why? If the TCP server were to support *n* simultaneous connections, each from a different client host, how many sockets would the TCP server need?

R27. For the client-server application over TCP described in **Section 2.7** , why must the server program be executed before the client program? For the client-server application over UDP, why may the client program be executed before the server program?

### Problems

P1. True or false?

1. A user requests a Web page that consists of some text and three images. For this page, the client will send one request message and receive four response messages.
2. Two distinct Web pages (for example, [***www.mit.edu/research.html***](http://www.mit.edu/research.html) and *www.mit.edu/students.html*) can be sent over the same persistent connection.
3. With nonpersistent connections between browser and origin server, it is possible for a single TCP segment to carry two distinct HTTP request messages.
4. The *Date:* header in the HTTP response message indicates when the object in the response was last modified.
5. HTTP response messages never have an empty message body.

P2. SMS, iMessage, and WhatsApp are all smartphone real-time messaging systems. After doing some research on the Internet, for each of these systems write one paragraph about the protocols they use. Then write a paragraph explaining how they differ.

P3. Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario?

P4. Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message).

The characters *<cr><lf>* are carriage return and line-feed characters (that is, the italized character string *<cr>* in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

*GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai*

*a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (*

*Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec*

*ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex*

*t/xml, application/xml, application/xhtml+xml, text /html;q=0.9, text/plain;q=0.8, image/png,\*/\*;q=0.5*

*<cr><lf>Accept-Language: en-us, en;q=0.5<cr><lf>AcceptEncoding: zip, deflate<cr><lf>Accept-Charset: ISO*

*-8859-1, utf-8;q=0.7,\*;q=0.7<cr><lf>Keep-Alive: 300<cr>*

*<lf>Connection:keep-alive<cr><lf><cr><lf>*

1. What is the URL of the document requested by the browser?
2. What version of HTTP is the browser running?
3. Does the browser request a non-persistent or a persistent connection?
4. What is the IP address of the host on which the browser is running?
5. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

P5. The text below shows the reply sent from the server in response to the HTTP GET message in the question above. Answer the following questions, indicating where in the message below you find the answer.

*HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008*

*12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)*

*<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46*

*GMT<cr><lf>ETag: ”526c3-f22-a88a4c80”<cr><lf>AcceptRanges: bytes<cr><lf>Content-Length: 3874<cr><lf> Keep-Alive: timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-Type: text/html; charset=*

*ISO-8859-1<cr><lf><cr><lf><!doctype html public ”-*

*//w3c//dtd html 4.0 transitional//en”><lf><html><lf> <head><lf> <meta http-equiv=”Content-Type” content=”text/html; charset=iso-8859-1”><lf> <meta name=”GENERATOR” content=”Mozilla/4.79 [en] (Windows NT 5.0; U) Netscape]”><lf> <title>CMPSCI 453 / 591 / NTU-ST550ASpring 2005 homepage</title><lf></head><lf>*

*<much more document text following here (not shown)>*

1. Was the server able to successfully find the document or not? What time was the document reply provided?
2. When was the document last modified?
3. How many bytes are there in the document being returned?
4. What are the first 5 bytes of the document being returned? Did the server agree to a persistent connection?

P6. Obtain the HTTP/1.1 specification (RFC 2616). Answer the following questions:

1. Explain the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed. Can the client, the server, or both signal the close of a connection?
2. What encryption services are provided by HTTP?
3. Can a client open three or more simultaneous connections with a given server?
4. Either a server or a client may close a transport connection between them if either one detects the connection has been idle for some time. Is it possible that one side starts closing a connection while the other side is transmitting data via this connection?

Explain.

P7. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that *n* DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT1,. . .,RTTn. Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

P8. Referring to Problem P7, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with

1. Non-persistent HTTP with no parallel TCP connections?
2. Non-persistent HTTP with the browser configured for 5 parallel connections?
3. Persistent HTTP?

P9. Consider **Figure 2.12** , for which there is an institutional network connected to the Internet. Suppose that the average object size is 850,000 bits and that the average request rate from the institution’s browsers to the origin servers is 16 requests per second. Also suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an

HTTP request until it receives the response is three seconds on average (see Section **2.2.5**). Model the total average response time as the sum of the average access delay (that is, the delay from Internet router to institution router) and the average Internet delay. For the average access delay, use Δ/(1−Δβ), where Δ is the average time required to send an object over the access link and b is the arrival rate of objects to the access link.

1. Find the total average response time.
2. Now suppose a cache is installed in the institutional LAN. Suppose the miss rate is 0.4.

Find the total response time.

P10. Consider a short, 10-meter link, over which a sender can transmit at a rate of 150 bits/sec in both directions. Suppose that packets containing data are 100,000 bits long, and packets containing only control (e.g., ACK or handshaking) are 200 bits long. Assume that *N* parallel connections each get 1/*N* of the link bandwidth. Now consider the HTTP protocol, and suppose that each downloaded object is 100 Kbits long, and that the initial downloaded object contains 10 referenced objects from the same sender. Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do you expect significant gains over the non-persistent case? Justify and explain your answer.

P11. Consider the scenario introduced in the previous problem. Now suppose that the link is shared by Bob with four other users. Bob uses parallel instances of non-persistent HTTP, and the other four users use non-persistent HTTP without parallel downloads.

1. Do Bob’s parallel connections help him get Web pages more quickly? Why or why not?
2. If all five users open five parallel instances of non-persistent HTTP, then would Bob’s parallel connections still be beneficial? Why or why not?

P12. Write a simple TCP program for a server that accepts lines of input from a client and prints the lines onto the server’s standard output. (You can do this by modifying the TCPServer.py program in the text.) Compile and execute your program. On any other machine that contains a Web browser, set the proxy server in the browser to the host that is running your server program; also configure the port number appropriately. Your browser should now send its GET request messages to your server, and your server should display the messages on its standard output. Use this platform to determine whether your browser generates conditional GET messages for objects that are locally cached.

P13. What is the difference between *MAIL FROM*: in SMTP and *From*: in the mail message itself?

P14. How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain.

P15. Read RFC 5321 for SMTP. What does MTA stand for? Consider the following received spam e-mail (modified from a real spam e-mail). Assuming only the originator of this spam e-mail is malicious and all other hosts are honest, identify the malacious host that has generated this spam e-mail.

*From - Fri Nov 07 13:41:30 2008*

*Return-Path: <tennis5@pp33head.com>*

*Received: from barmail.cs.umass.edu (barmail.cs.umass.edu*

*[128.119.240.3])*

*by cs.umass.edu (8.13.1/8.12.6) for*

*<*

*hg@cs.umass.edu>; Fri, 7 Nov*

*2008 13:27:10 -0500*

*Received: from asusus-4b96 (localhost [127.0.0.1]) by*

*barmail.cs.umass.edu (Spam Firewall) for <hg@cs.umass.edu>; Fri, 7*

*Nov 2008 13:27:07 -0500 (EST)*

*Received: from asusus-4b96 ([58.88.21.177]) by barmail.cs.umass.edu*

*for <hg@cs.umass.edu>; Fri, 07 Nov 2008 13:27:07 -0500 (EST)*

*Received: from [58.88.21.177] by inbnd55.exchangeddd.com; Sat, 8*

*Nov 2008 01:27:07 +0700*

*From: ”Jonny” <tennis5@pp33head.com>*

*To: <hg@cs.umass.edu>*

*Subject: How to secure your savings*

P16. Read the POP3 RFC, RFC 1939. What is the purpose of the UIDL POP3 command?

P17. Consider accessing your e-mail with POP3.

1. Suppose you have configured your POP mail client to operate in the download-anddelete mode. Complete the following transaction:

***C:***

*list*

***S:***

*1 498*

***S:***

*2 912*

***S:***

*.*

***C:***

*retr 1*

***S:***

*blah blah ...*

***S:***

*..........blah*

***S:***

*.*

*?*

*?*

1. Suppose you have configured your POP mail client to operate in the download-and-keep mode. Complete the following transaction:

***C:***

*list*

***S:***

*1 498*

***S:***

*2 912*

***S:***

*.*

***C:***

*retr 1*

***S:***

*blah blah ...*

***S:***

*..........blah*

***S:***

*.*

*?*

*?*

1. Suppose you have configured your POP mail client to operate in the download-and-keep mode. Using your transcript in part (b), suppose you retrieve messages 1 and 2, exit POP, and then five minutes later you again access POP to retrieve new e-mail. Suppose that in the five-minute interval no new messages have been sent to you. Provide a transcript of this second POP session.

P18.

1. What is a *whois* database?
2. Use various whois databases on the Internet to obtain the names of two DNS servers. Indicate which whois databases you used.
3. Use nslookup on your local host to send DNS queries to three DNS servers: your local DNS server and the two DNS servers you found in part (b). Try querying for Type A, NS, and MX reports. Summarize your findings.
4. Use nslookup to find a Web server that has multiple IP addresses. Does the Web server of your institution (school or company) have multiple IP addresses?
5. Use the ARIN whois database to determine the IP address range used by your university.
6. Describe how an attacker can use whois databases and the nslookup tool to perform reconnaissance on an institution before launching an attack.
7. Discuss why whois databases should be publicly available.

P19. In this problem, we use the useful *dig* tool available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that in **Figure 2.19** , a DNS server in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower-level DNS server. First read the man page for *dig*, and then answer the following questions.

1. Starting with a root DNS server (from one of the root servers [a-m].root-servers.net), initiate a sequence of queries for the IP address for your department’s Web server by using *dig.* Show the list of the names of DNS servers in the delegation chain in answering your query.
2. Repeat part (a) for several popular Web sites, such as [**google.com**](http://google.com/), [**yahoo.com**](http://yahoo.com/), or [**amazon.com**.](http://amazon.com/)

P20. Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

P21. Suppose that your department has a local DNS server for all computers in the department.

You are an ordinary user (i.e., not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

P22. Consider distributing a file of F=15 Gbits to *N* peers. The server has an upload rate of us=30 Mbps, and each peer has a download rate of di=2 Mbps and an upload rate of *u*. For N=10, 100, and 1,000 and u=300 Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combinations of *N* and *u* for both client-server distribution and P2P distribution.

P23. Consider distributing a file of *F* bits to *N* peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each

peer at different rates, as long as the combined rate does not exceed *us*.

1. Suppose that us/N≤dmin. Specify a distribution scheme that has a distribution time of *NF*/*us*.
2. Suppose that us/N≥dmin. Specify a distribution scheme that has a distribution time of

*F*/*d*min.

1. Conclude that the minimum distribution time is in general given by max{NF/us, F/dmin}.

P24. Consider distributing a file of *F* bits to *N* peers using a P2P architecture. Assume a fluid model. For simplicity assume that dmin is very large, so that peer download bandwidth is never a bottleneck.

1. Suppose that us≤(us+u1+…+uN)/N. Specify a distribution scheme that has a distribution time of *F*/*us*.
2. Suppose that us≥(us+u1+…+uN)/N. Specify a distribution scheme that has a distribution time of NF/(us+u1+…+uN).
3. Conclude that the minimum distribution time is in general given by max{F/us, NF/(us+u1+…+uN)}.

P25. Consider an overlay network with *N* active peers, with each pair of peers having an active

TCP connection. Additionally, suppose that the TCP connections pass through a total of *M* routers. How many nodes and edges are there in the corresponding overlay network?

P26. Suppose Bob joins a BitTorrent torrent, but he does not want to upload any data to any other peers (so called free-riding).

1. Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is Bob’s claim possible? Why or why not?
2. Bob further claims that he can further make his “free-riding” more efficient by using a collection of multiple computers (with distinct IP addresses) in the computer lab in his department. How can he do that?

P27. Consider a DASH system for which there are *N* video versions (at *N* different rates and qualities) and *N* audio versions (at *N* different rates and qualities). Suppose we want to allow the player to choose at any time any of the *N* video versions and any of the *N* audio versions.

* 1. If we create files so that the audio is mixed in with the video, so server sends only one media stream at given time, how many files will the server need to store (each a different

URL)?

* 1. If the server instead sends the audio and video streams separately and has the client synchronize the streams, how many files will the server need to store?

P28. Install and compile the Python programs TCPClient and UDPClient on one host and TCPServer and UDPServer on another host.

* 1. Suppose you run TCPClient before you run TCPServer. What happens? Why?
  2. Suppose you run UDPClient before you run UDPServer. What happens? Why?
  3. What happens if you use different port numbers for the client and server sides?

P29. Suppose that in UDPClient.py, after we create the socket, we add the line:

*clientSocket.bind((’’, 5432))*

Will it become necessary to change UDPServer.py? What are the port numbers for the sockets in UDPClient and UDPServer? What were they before making this change?

P30. Can you configure your browser to open multiple simultaneous connections to a Web site? What are the advantages and disadvantages of having a large number of simultaneous TCP connections?

P31. We have seen that Internet TCP sockets treat the data being sent as a byte stream but UDP sockets recognize message boundaries. What are one advantage and one disadvantage of byte-oriented API versus having the API explicitly recognize and preserve application-defined message boundaries?

P32. What is the Apache Web server? How much does it cost? What functionality does it currently have? You may want to look at Wikipedia to answer this question.

*Socket Programming Assignments*

The Companion Website includes six socket programming assignments. The first four assignments are summarized below. The fifth assignment makes use of the ICMP protocol and is summarized at the end of **Chapter 5**. The sixth assignment employs multimedia protocols and is summarized at the end of

**Chapter 9**. It is highly recommended that students complete several, if not all, of these assignments. Students can find full details of these assignments, as well as important snippets of the Python code, at

the Web site [**www.pearsonhighered.com/cs-resources**](http://www.pearsonhighered.com/cs-resources).

*Assignment 1: Web Server*

In this assignment, you will develop a simple Web server in Python that is capable of processing only one request. Specifically, your Web server will (i) create a connection socket when contacted by a client (browser); (ii) receive the HTTP request from this connection; (iii) parse the request to determine the specific file being requested; (iv) get the requested file from the server’s file system; (v) create an HTTP response message consisting of the requested file preceded by header lines; and (vi) send the response over the TCP connection to the requesting browser. If a browser requests a file that is not present in your server, your server should return a “404 Not Found” error message.

In the Companion Website, we provide the skeleton code for your server. Your job is to complete the code, run your server, and then test your server by sending requests from browsers running on different hosts. If you run your server on a host that already has a Web server running on it, then you should use a different port than port 80 for your Web server.

*Assignment 2: UDP Pinger*

In this programming assignment, you will write a client ping program in Python. Your client will send a simple ping message to a server, receive a corresponding pong message back from the server, and determine the delay between when the client sent the ping message and received the pong message. This delay is called the Round Trip Time (RTT). The functionality provided by the client and server is similar to the functionality provided by standard ping program available in modern operating systems. However, standard ping programs use the Internet Control Message Protocol (ICMP) (which we will study in **Chapter 5**). Here we will create a nonstandard (but simple!) UDP-based ping program.

Your ping program is to send 10 ping messages to the target server over UDP. For each message, your client is to determine and print the RTT when the corresponding pong message is returned. Because UDP is an unreliable protocol, a packet sent by the client or server may be lost. For this reason, the client cannot wait indefinitely for a reply to a ping message. You should have the client wait up to one second for a reply from the server; if no reply is received, the client should assume that the packet was lost and print a message accordingly.

In this assignment, you will be given the complete code for the server (available in the Companion Website). Your job is to write the client code, which will be very similar to the server code. It is recommended that you first study carefully the server code. You can then write your client code, liberally cutting and pasting lines from the server code.

*Assignment 3: Mail Client*

The goal of this programming assignment is to create a simple mail client that sends e-mail to any recipient. Your client will need to establish a TCP connection with a mail server (e.g., a Google mail server), dialogue with the mail server using the SMTP protocol, send an e-mail message to a recipient (e.g., your friend) via the mail server, and finally close the TCP connection with the mail server.

For this assignment, the Companion Website provides the skeleton code for your client. Your job is to complete the code and test your client by sending e-mail to different user accounts. You may also try sending through different servers (for example, through a Google mail server and through your university mail server).

*Assignment 4: Multi-Threaded Web Proxy*

In this assignment, you will develop a Web proxy. When your proxy receives an HTTP request for an object from a browser, it generates a new HTTP request for the same object and sends it to the origin server. When the proxy receives the corresponding HTTP response with the object from the origin server, it creates a new HTTP response, including the object, and sends it to the client. This proxy will be multi-threaded, so that it will be able to handle multiple requests at the same time.

For this assignment, the Companion Website provides the skeleton code for the proxy server. Your job is to complete the code, and then test it by having different browsers request Web objects via your proxy.

## Chapter 3 Review Questions

*SECTIONS 3.1–3.3*

R1. Suppose the network layer provides the following service. The network layer in the source host accepts a segment of maximum size 1,200 bytes and a destination host address from the transport layer. The network layer then guarantees to deliver the segment to the transport layer at the destination host. Suppose many network application processes can be running at the destination host.

1. Design the simplest possible transport-layer protocol that will get application data to the desired process at the destination host. Assume the operating system in the destination host has assigned a 4-byte port number to each running application process.
2. Modify this protocol so that it provides a “return address” to the destination process.
3. In your protocols, does the transport layer “have to do anything” in the core of the computer network?

R2. Consider a planet where everyone belongs to a family of six, every family lives in its own house, each house has a unique address, and each person in a given house has a unique name. Suppose this planet has a mail service that delivers letters from source house to destination house. The mail service requires that (1) the letter be in an envelope, and that (2) the address of the destination house (and nothing more) be clearly written on the envelope. Suppose each family has a delegate family member who collects and distributes letters for the other family members. The letters do not necessarily provide any indication of the recipients of the letters.

1. Using the solution to Problem R1 above as inspiration, describe a protocol that the delegates can use to deliver letters from a sending family member to a receiving family member.
2. In your protocol, does the mail service ever have to open the envelope and examine the letter in order to provide its service?

R3. Consider a TCP connection between Host A and Host B. Suppose that the TCP segments traveling from Host A to Host B have source port number *x* and destination port number *y*. What are the source and destination port numbers for the segments traveling from Host B to Host A?

R4. Describe why an application developer might choose to run an application over UDP rather than TCP.

R5. Why is it that voice and video traffic is often sent over TCP rather than UDP in today’s

Internet? (*Hint*: The answer we are looking for has nothing to do with TCP’s congestion-control mechanism.)

R6. Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

R7. Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B each send a UDP segment to Host C with destination port number 6789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

R8. Suppose that a Web server runs in Host C on port 80. Suppose this Web server uses persistent connections, and is currently receiving requests from two different Hosts, A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain.

*SECTION 3.4*

R9. In our *rdt* protocols, why did we need to introduce sequence numbers?

R10. In our *rdt* protocols, why did we need to introduce timers?

R11. Suppose that the roundtrip delay between sender and receiver is constant and known to the sender. Would a timer still be necessary in protocol *rdt 3.0*, assuming that packets can be lost? Explain.

R12. Visit the Go-Back-N Java applet at the companion Web site.

1. Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation.

Describe what happens.

1. Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens.
2. Finally, try sending six packets. What happens?

R13. Repeat R12, but now with the Selective Repeat Java applet. How are Selective Repeat and Go-Back-N different?

*SECTION 3.5*

R14. True or false?

1. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.
2. The size of the TCP *rwnd* never changes throughout the duration of the connection.
3. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.
4. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is *m*, then the sequence number for the subsequent segment will necessarily be m+1.
5. The TCP segment has a field in its header for *rwnd*.
6. Suppose that the last *SampleRTT* in a TCP connection is equal to 1 sec. The current value of *TimeoutInterval* for the connection will necessarily be ≥1 sec.
7. Suppose Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgment number is necessarily 42.

R15. Suppose Host A sends two TCP segments back to back to Host B over a TCP connection.

The first segment has sequence number 90; the second has sequence number 110.

1. How much data is in the first segment?
2. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

R16. Consider the Telnet example discussed in **Section 3.5** . A few seconds after the user types the letter ‘C,’ the user types the letter ‘R.’ After typing the letter ‘R,’ how many segments are sent, and what is put in the sequence number and acknowledgment fields of the segments?

*SECTION 3.7*

R17. Suppose two TCP connections are present over some bottleneck link of rate *R* bps. Both connections have a huge file to send (in the same direction over the bottleneck link). The transmissions of the files start at the same time. What transmission rate would TCP like to give to each of the connections?

R18. True or false? Consider congestion control in TCP. When the timer expires at the sender, the value of *ssthresh* is set to one half of its previous value.

R19. In the discussion of TCP splitting in the sidebar in **Section 3.7** , it was claimed that the response time with TCP splitting is approximately 4⋅RTTFE+RTTBE+processing time. Justify this claim.

Problems

P1. Suppose Client A initiates a Telnet session with Server S. At about the same time, Client B

also initiates a Telnet session with Server S. Provide possible source and destination port numbers for

1. The segments sent from A to S.
2. The segments sent from B to S.
3. The segments sent from S to A.
4. The segments sent from S to B.
5. If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?
6. How about if they are the same host?

P2. Consider **Figure 3.5** . What are the source and destination port values in the segments flowing from the server back to the clients’ processes? What are the IP addresses in the network-layer datagrams carrying the transport-layer segments?

P3. UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

P4.

1. Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these 2 bytes?
2. Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1s complement of the sum of these 2 bytes?
3. For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1s complement doesn’t change.

P5. Suppose that the UDP receiver computes the Internet checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? Explain.

P6. Consider our motivation for correcting protocol *rdt2.1*. Show that the receiver, shown in

**Figure 3.57** , when operating with the sender shown in **Figure 3.11** , can lead the sender and receiver to enter into a deadlock state, where each is waiting for an event that will never occur.

P7. In protocol *rdt3.0*, the ACK packets flowing from the receiver to the sender do not have sequence numbers (although they do have an ACK field that contains the sequence number of the packet they are acknowledging). Why is it that our ACK packets do not require sequence numbers?

A diagram of a diagram

AI-generated content may be incorrect.

*Figure 3.57 An incorrect receiver for protocol rdt 2.1*

P8. Draw the FSM for the receiver side of protocol *rdt3.0*.

P9. Give a trace of the operation of protocol *rdt3.0* when data packets and acknowledgment packets are garbled. Your trace should be similar to that used in **Figure 3.16** .

P10. Consider a channel that can lose packets but has a maximum delay that is known. Modify protocol *rdt2.1* to include sender timeout and retransmit. Informally argue why your protocol can communicate correctly over this channel.

P11. Consider the *rdt2.2* receiver in **Figure 3.14** , and the creation of a new packet in the self-transition (i.e., the transition from the state back to itself) in the Wait-for-0-from-below and

the Wait-for-1-from-below states: *sndpkt=make\_pkt(ACK, 1, checksum)* and *sndpkt=make\_pkt(ACK, 0, checksum)*. Would the protocol work correctly if this action

were removed from the self-transition in the Wait-for-1-from-below state? Justify your answer.

What if this event were removed from the self-transition in the Wait-for-0-from-below state? [*Hint*: In this latter case, consider what would happen if the first sender-to-receiver packet were corrupted.]

P12. The sender side of *rdt3.0* simply ignores (that is, takes no action on) all received packets that are either in error or have the wrong value in the *acknum* field of an acknowledgment packet. Suppose that in such circumstances, *rdt3.0* were simply to retransmit the current data packet. Would the protocol still work? (*Hint*: Consider what would happen if there were only bit errors; there are no packet losses but premature timeouts can occur. Consider how many times the *n*th packet is sent, in the limit as *n* approaches infinity.) P13. Consider the *rdt* *3.0* protocol. Draw a diagram showing that if the network connection between the sender and receiver can reorder messages (that is, that two messages propagating in the medium between the sender and receiver can be reordered), then the alternating-bit protocol will not work correctly (make sure you clearly identify the sense in which it will not work correctly). Your diagram should have the sender on the left and the receiver on the right, with the time axis running down the page, showing data (D) and acknowledgment (A) message exchange. Make sure you indicate the sequence number associated with any data or acknowledgment segment.

P14. Consider a reliable data transfer protocol that uses only negative acknowledgments. Suppose the sender sends data only infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why? Now suppose the sender has a lot of data to send and the endto-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

P15. Consider the cross-country example shown in **Figure 3.17** . How big would the window size have to be for the channel utilization to be greater than 98 percent? Suppose that the size of a packet is 1,500 bytes, including both header fields and data.

P16. Suppose an application uses *rdt 3.0* as its transport layer protocol. As the stop-and-wait protocol has very low channel utilization (shown in the cross-country example), the designers of this application let the receiver keep sending back a number (more than two) of alternating ACK 0 and ACK 1 even if the corresponding data have not arrived at the receiver. Would this application design increase the channel utilization? Why? Are there any potential problems with this approach? Explain.

P17. Consider two network entities, A and B, which are connected by a perfect bi-directional channel (i.e., any message sent will be received correctly; the channel will not corrupt, lose, or re-order packets). A and B are to deliver data messages to each other in an alternating manner: First, A must deliver a message to B, then B must deliver a message to A, then A must deliver a message to B and so on. If an entity is in a state where it should not attempt to deliver a message to the other side, and there is an event like *rdt\_send(data)* call from above that attempts to pass data down for transmission to the other side, this call from above can simply be ignored with a call to *rdt\_unable\_to\_send(data)*, which informs the higher layer that it is currently not able to send data. [Note: This simplifying assumption is made so you don’t have to worry about buffering data.]

Draw a FSM specification for this protocol (one FSM for A, and one FSM for B!). Note that you do not have to worry about a reliability mechanism here; the main point of this question is to create a FSM specification that reflects the synchronized behavior of the two entities. You should use the following events and actions that have the same meaning as protocol rdt1.0 in **Figure**

**3.9** : *rdt\_send(data), packet = make\_pkt(data)*, *udt\_send(packet), rdt\_rcv(packet)*, *extract (packet, data), deliver\_data(data)*. Make sure your

protocol reflects the strict alternation of sending between A and B. Also, make sure to indicate the initial states for A and B in your FSM descriptions.

P18. In the generic SR protocol that we studied in **Section 3.4.4** , the sender transmits a message as soon as it is available (if it is in the window) without waiting for an acknowledgment. Suppose now that we want an SR protocol that sends messages two at a time. That is, the sender will send a pair of messages and will send the next pair of messages only when it knows that both messages in the first pair have been received correctly.

Suppose that the channel may lose messages but will not corrupt or reorder messages. Design an error-control protocol for the unidirectional reliable transfer of messages. Give an FSM description of the sender and receiver. Describe the format of the packets sent between sender and receiver, and vice versa. If you use any procedure calls other than those in **Section 3.4** (for example, *udt\_send()*, *start\_timer()*, *rdt\_rcv()*, and so on), clearly state their actions. Give an example (a timeline trace of sender and receiver) showing how your protocol recovers from a lost packet.

P19. Consider a scenario in which Host A wants to simultaneously send packets to Hosts B and C. A is connected to B and C via a broadcast channel—a packet sent by A is carried by the channel to both B and C. Suppose that the broadcast channel connecting A, B, and C can independently lose and corrupt packets (and so, for example, a packet sent from A might be correctly received by B, but not by C). Design a stop-and-wait-like error-control protocol for reliably transferring packets from A to B and C, such that A will not get new data from the upper layer until it knows that both B and C have correctly received the current packet. Give FSM descriptions of A and C. (*Hint:* The FSM for B should be essentially the same as for C.) Also, give a description of the packet format(s) used.

P20. Consider a scenario in which Host A and Host B want to send messages to Host C. Hosts A and C are connected by a channel that can lose and corrupt (but not reorder) messages.

Hosts B and C are connected by another channel (independent of the channel connecting A and C) with the same properties. The transport layer at Host C should alternate in delivering messages from A and B to the layer above (that is, it should first deliver the data from a packet from A, then the data from a packet from B, and so on). Design a stop-and-wait-like error-control protocol for reliably transferring packets from A and B to C, with alternating delivery at C as described above. Give FSM descriptions of A and C. (*Hint:* The FSM for B should be essentially the same as for A.) Also, give a description of the packet format(s) used.

P21. Suppose we have two network entities, A and B. B has a supply of data messages that will be sent to A according to the following conventions. When A gets a request from the layer above to get the next data (D) message from B, A must send a request (R) message to B on the A-to-B channel. Only when B receives an R message can it send a data (D) message back to A on the B-to-A channel. A should deliver exactly one copy of each D message to the layer above. R messages can be lost (but not corrupted) in the A-to-B channel; D messages, once sent, are always delivered correctly. The delay along both channels is unknown and variable.

Design (give an FSM description of) a protocol that incorporates the appropriate mechanisms to compensate for the loss-prone A-to-B channel and implements message passing to the layer above at entity A, as discussed above. Use only those mechanisms that are absolutely necessary.

P22. Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1,024. Suppose that at time *t*, the next in-order packet that the receiver is expecting has a sequence number of *k*. Assume that the medium does not reorder messages. Answer the following questions:

1. What are the possible sets of sequence numbers inside the sender’s window at time *t*?

Justify your answer.

1. What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time *t*? Justify your answer.

P23. Consider the GBN and SR protocols. Suppose the sequence number space is of size *k*. What is the largest allowable sender window that will avoid the occurrence of problems such as that in **Figure 3.27** for each of these protocols?

P24. Answer true or false to the following questions and briefly justify your answer:

1. With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
2. With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
3. The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.
4. The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

P25. We have said that an application may choose UDP for a transport protocol because UDP offers finer application control (than TCP) of what data is sent in a segment and when.

1. Why does an application have more control of what data is sent in a segment?
2. Why does an application have more control on when the segment is sent?

P26. Consider transferring an enormous file of *L* bytes from Host A to Host B. Assume an MSS of 536 bytes.

1. What is the maximum value of *L* such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.
2. For the *L* you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

P27. Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B backto-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

1. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?
2. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?
3. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?
4. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

P28. Host A and B are directly connected with a 100 Mbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection.

Host A can send its application data into its TCP socket at a rate as high as 120 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 50 Mbps. Describe the effect of TCP flow control.

P29. SYN cookies were discussed in **Section 3.5.6** .

1. Why is it necessary for the server to use a special initial sequence number in the

SYNACK?

1. Suppose an attacker knows that a target host uses SYN cookies. Can the attacker create half-open or fully open connections by simply sending an ACK packet to the target? Why or why not?
2. Suppose an attacker collects a large amount of initial sequence numbers sent by the server. Can the attacker cause the server to create many fully open connections by sending ACKs with those initial sequence numbers? Why?

P30. Consider the network shown in Scenario 2 in **Section 3.6.1** . Suppose both sending hosts A and B have some fixed timeout values.

1. Argue that increasing the size of the finite buffer of the router might possibly decrease the throughput (λ*out*).
2. Now suppose both hosts dynamically adjust their timeout values (like what TCP does) based on the buffering delay at the router. Would increasing the buffer size help to increase the throughput? Why?

P31. Suppose that the five measured *SampleRTT* values (see **Section 3.5.3** ) are 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Compute the *EstimatedRTT* after each of these SampleRTT values is obtained, using a value of α=0.125 and assuming that the value of *EstimatedRTT* was 100 ms just before the first of these five samples were obtained. Compute also the *DevRTT* after each sample is obtained, assuming a value of β=0.25 and assuming the value of *DevRTT* was 5 ms just before the first of these five samples was obtained. Last, compute the TCP

*TimeoutInterval* after each of these samples is obtained.

P32. Consider the TCP procedure for estimating RTT. Suppose that α=0.1. Let *SampleRTT***1** be the most recent sample RTT, let *SampleRTT***2** be the next most recent sample RTT, and so on.

1. For a given TCP connection, suppose four acknowledgments have been returned with corresponding sample RTTs: *SampleRTT***4**, *SampleRTT***3**, *SampleRTT***2**, and *SampleRTT***1**. Express *EstimatedRTT* in terms of the four sample RTTs.
2. Generalize your formula for *n* sample RTTs.
3. For the formula in part (b) let *n* approach infinity. Comment on why this averaging procedure is called an exponential moving average.

P33. In **Section 3.5.3** , we discussed TCP’s estimation of RTT. Why do you think TCP avoids measuring the *SampleRTT* for retransmitted segments?

P34. What is the relationship between the variable *SendBase* in **Section 3.5.4** and the variable *LastByteRcvd* in **Section 3.5.5** ?

P35. What is the relationship between the variable *LastByteRcvd* in **Section 3.5.5** and the variable y in Section **3.5.4**?

P36. In **Section 3.5.4** , we saw that TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first duplicate ACK for a segment is received?

P37. Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 data segments have been correctly received by Host B.

1. How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.
2. If the timeout values for all three protocol are much longer than 5 RTT, then which protocol successfully delivers all five data segments in shortest time interval?

P38. In our description of TCP in **Figure 3.53** , the value of the threshold, *ssthresh*, is set as *ssthresh=cwnd/2* in several places and *ssthresh* value is referred to as being set to half

the window size when a loss event occurred. Must the rate at which the sender is sending when the loss event occurred be approximately equal to *cwnd* segments per RTT? Explain your answer. If your answer is no, can you suggest a different manner in which *ssthresh* should be set?

P39. Consider **Figure 3.46(b)** . If λ′in increases beyond *R*/2, can λout increase beyond *R*/3?

Explain. Now consider **Figure 3.46(c)** . If λ′in increases beyond *R*/2, can λout increase beyond R/4 under the assumption that a packet will be forwarded twice on average from the router to the receiver? Explain.

P40. Consider **Figure 3.58** . Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.



Examining the behavior of TCP

1. Identify the intervals of time when TCP slow start is operating.
2. Identify the intervals of time when TCP congestion avoidance is operating.
3. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
4. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

A graph with blue lines and dots

AI-generated content may be incorrect.

*Figure 3.58 TCP window size as a function of time*

1. What is the initial value of *ssthresh* at the first transmission round?
2. What is the value of *ssthresh* at the 18th transmission round?
3. What is the value of *ssthresh* at the 24th transmission round?
4. During what transmission round is the 70th segment sent?
5. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of

*ssthresh*?

1. Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the *ssthresh* and the congestion window size at the 19th round?
2. Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

P41. Refer to **Figure 3.55** , which illustrates the convergence of TCP’s AIMD algorithm. Suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount. Would the resulting AIAD algorithm converge to an equal share algorithm? Justify your answer using a diagram similar to **Figure 3.55** .

P42. In **Section 3.5.4** , we discussed the doubling of the timeout interval after a timeout event. This mechanism is a form of congestion control. Why does TCP need a window-based congestion-control mechanism (as studied in **Section 3.7** ) in addition to this doubling-timeoutinterval mechanism?

P43. Host A is sending an enormous file to Host B over a TCP connection. Over this connection there is never any packet loss and the timers never expire. Denote the transmission rate of the link connecting Host A to the Internet by *R* bps. Suppose that the process in Host A is capable of sending data into its TCP socket at a rate *S* bps, where S=10⋅R. Further suppose that the TCP receive buffer is large enough to hold the entire file, and the send buffer can hold only one percent of the file. What would prevent the process in Host A from continuously passing data to its TCP socket at rate *S* bps? TCP flow control? TCP congestion control? Or something else?

Elaborate.

P44. Consider sending a large file from a host to another over a TCP connection that has no loss.

1. Suppose TCP uses AIMD for its congestion control without slow start. Assuming *cwnd* increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for *cwnd* increase from 6 MSS to 12 MSS (assuming no loss events)?
2. What is the average throughout (in terms of MSS and RTT) for this connection up through time=6 RTT?

P45. Recall the macroscopic description of TCP throughput. In the period of time from when the connection’s rate varies from *W*/(2 · RTT) to *W/RTT*, only one packet is lost (at the very end of the period).

1. Show that the loss rate (fraction of packets lost) is equal to

L=loss rate=138W2+34W

1. Use the result above to show that if a connection has loss rate *L*, then its average rate is approximately given by

≈1.22⋅MSSRTTL

P46. Consider that only a single TCP (Reno) connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver’s receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion avoidance phase, that is, ignore slow start.

1. What is the maximum window size (in segments) that this TCP connection can achieve?
2. What is the average window size (in segments) and average throughput (in bps) of this TCP connection?
3. How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

P47. Consider the scenario described in the previous problem. Suppose that the 10Mbps link can buffer a finite number of segments. Argue that in order for the link to always be busy sending data, we would like to choose a buffer size that is at least the product of the link speed *C* and the two-way propagation delay between the sender and the receiver.

P48. Repeat Problem 46, but replacing the 10 Mbps link with a 10 Gbps link. Note that in your answer to part c, you will realize that it takes a very long time for the congestion window size to reach its maximum window size after recovering from a packet loss. Sketch a solution to solve this problem.

P49. Let *T* (measured by RTT) denote the time interval that a TCP connection takes to increase its congestion window size from *W/2* to *W*, where *W* is the maximum congestion window size.

Argue that T is a function of TCP’s average throughput.

P50. Consider a simplified TCP’s AIMD algorithm where the congestion window size is measured in number of segments, not in bytes. In additive increase, the congestion window size increases by one segment in each RTT. In multiplicative decrease, the congestion window size decreases by half (if the result is not an integer, round down to the nearest integer). Suppose that two TCP connections, C1 and C2, share a single congested link of speed 30 segments per second. Assume that both C1 and C2 are in the congestion avoidance phase. Connection C1’s

RTT is 50 msec and connection C2’s RTT is 100 msec. Assume that when the data rate in the link exceeds the link’s speed, all TCP connections experience data segment loss.

1. If both C1 and C2 at time t0 have a congestion window of 10 segments, what are their congestion window sizes after 1000 msec?
2. In the long run, will these two connections get the same share of the bandwidth of the congested link? Explain.

P51. Consider the network described in the previous problem. Now suppose that the two TCP connections, C1 and C2, have the same RTT of 100 msec. Suppose that at time t0, C1’s congestion window size is 15 segments but C2’s congestion window size is 10 segments.

1. What are their congestion window sizes after 2200 msec?
2. In the long run, will these two connections get about the same share of the bandwidth of the congested link?
3. We say that two connections are synchronized, if both connections reach their maximum window sizes at the same time and reach their minimum window sizes at the same time. In the long run, will these two connections get synchronized eventually? If so, what are their maximum window sizes?
4. Will this synchronization help to improve the utilization of the shared link? Why? Sketch some idea to break this synchronization.

P52. Consider a modification to TCP’s congestion control algorithm. Instead of additive increase, we can use multiplicative increase. A TCP sender increases its window size by a small positive constant a(0<a<1) whenever it receives a valid ACK. Find the functional relationship between loss rate L and maximum congestion window W. Argue that for this modified TCP, regardless of TCP’s average throughput, a TCP connection always spends the same amount of time to increase its congestion window size from *W/2* to *W*.

P53. In our discussion of TCP futures in **Section 3.7** , we noted that to achieve a throughput of 10 Gbps, TCP could only tolerate a segment loss probability of 2⋅10−10 (or equivalently, one loss event for every 5,000,000,000 segments). Show the derivation for the values of 2⋅10−10 (1 out of 5,000,000) for the RTT and MSS values given in **Section 3.7** . If TCP needed to support a 100 Gbps connection, what would the tolerable loss be?

P54. In our discussion of TCP congestion control in **Section 3.7** , we implicitly assumed that the TCP sender always had data to send. Consider now the case that the TCP sender sends a large amount of data and then goes idle (since it has no more data to send) at *t*1. TCP remains idle for a relatively long period of time and then wants to send more data at *t*2. What are the advantages and disadvantages of having TCP use the *cwnd* and *ssthresh* values from *t*1 when starting to send data at *t*2? What alternative would you recommend? Why?

P55. In this problem we investigate whether either UDP or TCP provides a degree of end-point authentication.

1. Consider a server that receives a request within a UDP packet and responds to that request within a UDP packet (for example, as done by a DNS server). If a client with IP address X spoofs its address with address Y, where will the server send its response?
2. Suppose a server receives a SYN with IP source address Y, and after responding with a SYNACK, receives an ACK with IP source address Y with the correct acknowledgment number. Assuming the server chooses a random initial sequence number and there is no “man-in-the-middle,” can the server be certain that the client is indeed at Y (and not at some other address X that is spoofing Y)?

P56. In this problem, we consider the delay introduced by the TCP slow-start phase. Consider a client and a Web server directly connected by one link of rate *R*. Suppose the client wants to retrieve an object whose size is exactly equal to 15 *S*, where *S* is the maximum segment size (MSS). Denote the round-trip time between client and server as RTT (assumed to be constant). Ignoring protocol headers, determine the time to retrieve the object (including TCP connection establishment) when

* 1. 4 S/R>S/R+RTT>2S/R
  2. S/R+RTT>4 S/R
  3. S/R>RTT.

## Chapter 4 Review Questions

*SECTION 4.1*

R1. Let’s review some of the terminology used in this textbook. Recall that the name of a transport-layer packet is *segment* and that the name of a link-layer packet is *frame*. What is the name of a network-layer packet? Recall that both routers and link-layer switches are called *packet switches*. What is the fundamental difference between a router and link-layer switch?

R2. We noted that network layer functionality can be broadly divided into data plane functionality and control plane functionality. What are the main functions of the data plane? Of the control plane?

R3. We made a distinction between the forwarding function and the routing function performed in the network layer. What are the key differences between routing and forwarding?

R4. What is the role of the forwarding table within a router?

R5. We said that a network layer’s service model “defines the characteristics of end-to-end transport of packets between sending and receiving hosts.” What is the service model of the Internet’s network layer? What guarantees are made by the Internet’s service model regarding the host-to-host delivery of datagrams?

*SECTION 4.2*

R6. In **Section 4.2** , we saw that a router typically consists of input ports, output ports, a switching fabric and a routing processor. Which of these are implemented in hardware and which are implemented in software? Why? Returning to the notion of the network layer’s data plane and control plane, which are implemented in hardware and which are implemented in software?

Why?

R7. Discuss why each input port in a high-speed router stores a shadow copy of the forwarding table.

R8. What is meant by destination-based forwarding? How does this differ from generalized forwarding (assuming you’ve read **Section 4.4** , which of the two approaches are adopted by Software-Defined Networking)?

R9. Suppose that an arriving packet matches two or more entries in a router’s forwarding table.

With traditional destination-based forwarding, what rule does a router apply to determine which

of these rules should be applied to determine the output port to which the arriving packet should be switched?

R10. Three types of switching fabrics are discussed in **Section 4.2** . List and briefly describe each type. Which, if any, can send multiple packets across the fabric in parallel?

R11. Describe how packet loss can occur at input ports. Describe how packet loss at input ports can be eliminated (without using infinite buffers).

R12. Describe how packet loss can occur at output ports. Can this loss be prevented by increasing the switch fabric speed?

R13. What is HOL blocking? Does it occur in input ports or output ports?

R14. In **Section 4.2** , we studied FIFO, Priority, Round Robin (RR), and Weighted Fair Queueing (WFQ) packet scheduling disciplines? Which of these queueing disciplines ensure that all packets depart in the order in which they arrived?

R15. Give an example showing why a network operator might want one class of packets to be given priority over another class of packets.

R16. What is an essential different between RR and WFQ packet scheduling? Is there a case (Hint: Consider the WFQ weights) where RR and WFQ will behave exactly the same?

*SECTION 4.3*

R17. Suppose Host A sends Host B a TCP segment encapsulated in an IP datagram. When Host B receives the datagram, how does the network layer in Host B know it should pass the segment (that is, the payload of the datagram) to TCP rather than to UDP or to some other upper-layer protocol?

R18. What field in the IP header can be used to ensure that a packet is forwarded through no more than *N* routers?

R19. Recall that we saw the Internet checksum being used in both transport-layer segment (in UDP and TCP headers, **Figures 3.7** and **3.29** respectively) and in network-layer datagrams (IP header, **Figure 4.16** ). Now consider a transport layer segment encapsulated in an IP datagram. Are the checksums in the segment header and datagram header computed over any common bytes in the IP datagram? Explain your answer.

R20. When a large datagram is fragmented into multiple smaller datagrams, where are these smaller datagrams reassembled into a single larger datagram?

R21. Do routers have IP addresses? If so, how many?

R22. What is the 32-bit binary equivalent of the IP address 223.1.3.27?

R23. Visit a host that uses DHCP to obtain its IP address, network mask, default router, and IP address of its local DNS server. List these values.

R24. Suppose there are three routers between a source host and a destination host. Ignoring fragmentation, an IP datagram sent from the source host to the destination host will travel over how many interfaces? How many forwarding tables will be indexed to move the datagram from the source to the ­destination?

R25. Suppose an application generates chunks of 40 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data?

R26. Suppose you purchase a wireless router and connect it to your cable modem. Also suppose that your ISP dynamically assigns your connected device (that is, your wireless router) one IP address. Also suppose that you have five PCs at home that use 802.11 to wirelessly connect to your wireless router. How are IP addresses assigned to the five PCs? Does the wireless router use NAT? Why or why not?

R27. What is meant by the term “route aggregation”? Why is it useful for a router to perform route aggregation?

R28. What is meant by a “plug-and-play” or “zeroconf” protocol?

R29. What is a private network address? Should a datagram with a private network address ever be present in the larger public Internet? Explain.

R30. Compare and contrast the IPv4 and the IPv6 header fields. Do they have any fields in common?

R31. It has been said that when IPv6 tunnels through IPv4 routers, IPv6 treats the IPv4 tunnels as link-layer protocols. Do you agree with this statement? Why or why not?

*SECTION 4.4*

R32. How does generalized forwarding differ from destination-based ­forwarding?

R33. What is the difference between a forwarding table that we encountered in destinationbased forwarding in **Section 4.1** and OpenFlow’s flow table that we encountered in **Section 4.4** ?

R34. What is meant by the “match plus action” operation of a router or switch? In the case of destination-based forwarding packet switch, what is matched and what is the action taken? In the case of an SDN, name three fields that can be matched, and three actions that can be taken.

R35. Name three header fields in an IP datagram that can be “matched” in OpenFlow 1.0 generalized forwarding. What are three IP datagram header fields that *cannot* be “matched” in OpenFlow?

Problems

P1. Consider the network below.

1. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.
2. Can you write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4? (Hint: This is a trick question.)

A diagram of a network

AI-generated content may be incorrect.

P2. Suppose two packets arrive to two different input ports of a router at exactly the same time. Also suppose there are no other packets anywhere in the router.

1. Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a shared bus?
2. Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses switching via memory?
3. Suppose the two packets are to be forwarded to the same output port. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a crossbar?

P3. In **Section 4.2** , we noted that the maximum queuing delay is *(n–1)D* if the switching fabric is *n* times faster than the input line rates. Suppose that all packets are of the same length, *n* packets arrive at the same time to the *n* input ports, and all *n* packets want to be forwarded to different output ports. What is the maximum delay for a packet for the (a) memory, (b) bus, and (c) crossbar switching fabrics?

P4. Consider the switch shown below. Suppose that all datagrams have the same fixed length, that the switch operates in a slotted, synchronous manner, and that in one time slot a datagram can be transferred from an input port to an output port. The switch fabric is a crossbar so that at most one datagram can be transferred to a given output port in a time slot, but different output ports can receive datagrams from different input ports in a single time slot. What is the minimal number of time slots needed to transfer the packets shown from input ports to their output ports, assuming any input queue scheduling order you want (i.e., it need not have HOL blocking)? What is the largest number of slots needed, assuming the worst-case scheduling order you can devise, assuming that a non-empty input queue is never idle?

A screenshot of a computer

AI-generated content may be incorrect.

P5. Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

|  |  |
| --- | --- |
| Destination Address Range | Link Interface |
| 11100000 00000000 00000000 00000000 through  11100000 00111111 11111111 11111111 | 0 |
| 11100000 01000000 00000000 00000000 through  11100000 01000000 11111111 11111111 | 1 |
| 1. 01000001 00000000 00000000through 2. 01111111 11111111 11111111 | 2 |
| otherwise | 3 |

1. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.
2. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

11001000 10010001 01010001 01010101

11100001 01000000 11000011 00111100

11100001 10000000 00010001 01110111

P6. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

|  |  |
| --- | --- |
| Prefix Match | Interface |
|  |  |
| 00 | 0 |
| 010 | 1 |
| 011 | 2 |
| 10 | 2 |
| 11 | 3 |

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

P7. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

|  |  |
| --- | --- |
| Prefix Match | Interface |
| 1 | 0 |
| 10 | 1 |
| 111 | 2 |
| otherwise | 3 |

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

P8. Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support at least 60 interfaces, Subnet 2 is to support at least 90 interfaces, and Subnet 3 is to support at least 12 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

P9. In **Section 4.2.2** an example forwarding table (using longest prefix matching) is given.

Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

P10. In Problem P5 you are asked to provide a forwarding table (using longest prefix matching).

Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

P11. Consider a subnet with prefix 128.119.40.128/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network. Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form

a.b.c.d/x) for the four subnets?

P12. Consider the topology shown in **Figure 4.20** . Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

1. Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support 250 interfaces; Subnet B should have enough addresses to support 120 interfaces; and Subnet C should have enough addresses to support 120 interfaces. Of course, subnets D, E and F should each be able to support two interfaces.

For each subnet, the assignment should take the form a.b.c.d/x or a.b.c.d/x – e.f.g.h/y.

1. Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.

P13. Use the whois service at the American Registry for Internet Numbers ([**http://www.arin.net/ whois**)](http://www.arin.net/whois) to determine the IP address blocks for three universities. Can the whois services be used to determine with certainty the geographical location of a specific IP address? Use [**www.maxmind.com**](http://www.maxmind.com/) to determine the locations of the Web servers at each of these universities.

P14. Consider sending a 2400-byte datagram into a link that has an MTU of 700 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

P15. Suppose datagrams are limited to 1,500 bytes (including header) between source Host A and destination Host B. Assuming a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain how you computed your answer.

P16. Consider the network setup in **Figure 4.25** . Suppose that the ISP instead assigns the router the address 24.34.112.235 and that the network address of the home network is 192.168.1/24.

1. Assign addresses to all interfaces in the home network.
2. Suppose each host has two ongoing TCP connections, all to port 80 at host

128.119.40.86. Provide the six corresponding entries in the NAT translation table.

P17. Suppose you are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a random number, and the identification numbers of the subsequent IP packets are sequentially assigned. Assume all IP packets generated by hosts behind the NAT are sent to the outside world.

1. Based on this observation, and assuming you can sniff all packets sent by the NAT to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.
2. If the identification numbers are not sequentially assigned but randomly assigned, would

your technique work? Justify your answer.

P18. In this problem we’ll explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without applicationspecific NAT configuration. If you have difficulty devising such a technique, discuss why.

P19. Consider the SDN OpenFlow network shown in **Figure 4.30** . Suppose that the desired forwarding behavior for datagrams arriving at s2 is as follows:

any datagrams arriving on input port 1 from hosts h5 or h6 that are destined to hosts h1 or h2 should be forwarded over output port 2; any datagrams arriving on input port 2 from hosts h1 or h2 that are destined to hosts h5 or h6 should be forwarded over output port 1; any arriving datagrams on input ports 1 or 2 and destined to hosts h3 or h4 should be delivered to the host specified;

hosts h3 and h4 should be able to send datagrams to each other.

Specify the flow table entries in s2 that implement this forwarding behavior.

P20. Consider again the SDN OpenFlow network shown in **Figure 4.30** . Suppose that the desired forwarding behavior for datagrams arriving from hosts h3 or h4 at s2 is as follows:

any datagrams arriving from host h3 and destined for h1, h2, h5 or h6 should be forwarded in a clockwise direction in the network; any datagrams arriving from host h4 and destined for h1, h2, h5 or h6 should be forwarded in a counter-clockwise direction in the network.

Specify the flow table entries in s2 that implement this forwarding behavior.

P21. Consider again the scenario from P19 above. Give the flow tables entries at packet switches s1 and s3, such that any arriving datagrams with a source address of h3 or h4 are routed to the destination hosts specified in the destination address field in the IP datagram. (Hint: Your forwarding table rules should include the cases that an arriving datagram is destined for a directly attached host or should be forwarded to a neighboring router for eventual host delivery there.)

P22. Consider again the SDN OpenFlow network shown in **Figure 4.30** . Suppose we want switch s2 to function as a firewall. Specify the flow table in s2 that implements the following firewall behaviors (specify a different flow table for each of the four firewalling behaviors below) for delivery of datagrams destined to h3 and h4. You do not need to specify the forwarding behavior in s2 that forwards traffic to other routers.

Only traffic arriving from hosts h1 and h6 should be delivered to hosts h3 or h4 (i.e., that arriving traffic from hosts h2 and h5 is blocked).

Only TCP traffic is allowed to be delivered to hosts h3 or h4 (i.e., that UDP traffic is blocked).

Only traffic destined to h3 is to be delivered (i.e., all traffic to h4 is blocked).

Only UDP traffic from h1 and destined to h3 is to be delivered. All other traffic is blocked.

*Wireshark Lab*

In the Web site for this textbook, [**www.pearsonhighered.com/cs-resources**](http://www.pearsonhighered.com/cs-resources), you’ll find a Wireshark lab assignment that examines the operation of the IP protocol, and the IP datagram format in particular.

Vinton G. Cerf is Vice President and Chief Internet Evangelist for Google. He served for over 16 years at MCI in various positions, ending up his tenure there as Senior Vice President for Technology Strategy. He is widely known as the co-designer of the TCP/IP protocols and the architecture of the Internet. During his time from 1976 to 1982 at the US Department of Defense

Advanced Research Projects Agency (DARPA), he played a key role leading the development of

Internet and Internet-related data packet and security techniques. He received the US Presidential Medal of Freedom in 2005 and the US National Medal of Technology in 1997. He holds a BS in Mathematics from Stanford University and an MS and PhD in computer science from UCLA.

What brought you to specialize in networking?

I was working as a programmer at UCLA in the late 1960s. My job was supported by the US Defense Advanced Research Projects Agency (called ARPA then, called DARPA now). I was working in the laboratory of Professor Leonard Kleinrock on the Network Measurement Center of the newly created ARPAnet. The first node of the ARPAnet was installed at UCLA on September 1, 1969. I was responsible for programming a computer that was used to capture performance information about the ARPAnet and to report this information back for comparison with mathematical models and predictions of the performance of the network.

AN INTERVIEW WITH…

**Vinton G. Cerf**



Several of the other graduate students and I were made responsible for working on the so-called host-level protocols of the ARPAnet—the procedures and formats that would allow many different kinds of computers on the network to interact with each other. It was a fascinating exploration into a new world (for me) of distributed computing and communication.

Did you imagine that IP would become as pervasive as it is today when you first designed the

protocol?

When Bob Kahn and I first worked on this in 1973, I think we were mostly very focused on the central question: How can we make heterogeneous packet networks interoperate with one another, assuming we cannot actually change the networks themselves? We hoped that we could find a way to permit an arbitrary collection of packet-switched networks to be interconnected in a transparent fashion, so that host computers could communicate end-to-end without having to do any translations in between. I think we knew that we were dealing with powerful and expandable technology, but I doubt we had a clear image of what the world would be like with hundreds of millions of computers all interlinked on the Internet. What do you now envision for the future of networking and the Internet? What major

challenges/obstacles do you think lie ahead in their development?

I believe the Internet itself and networks in general will continue to proliferate. Already there is convincing evidence that there will be billions of Internet-enabled devices on the Internet, including appliances like cell phones, refrigerators, personal digital assistants, home servers, televisions, as well as the usual array of laptops, servers, and so on. Big challenges include support for mobility, battery life, capacity of the access links to the network, and ability to scale the optical core of the network up in an unlimited fashion. Designing an interplanetary extension of the Internet is a project in which I am deeply engaged at the Jet Propulsion Laboratory. We will need to cut over from IPv4 [32-bit addresses] to IPv6 [128 bits]. The list is long!

Who has inspired you professionally?

My colleague Bob Kahn; my thesis advisor, Gerald Estrin; my best friend, Steve Crocker (we met in high school and he introduced me to computers in 1960!); and the thousands of engineers who continue to evolve the Internet today.

Do you have any advice for students entering the networking/Internet field?

Think outside the limitations of existing systems—imagine what might be possible; but then do the hard work of figuring out how to get there from the current state of affairs. Dare to dream: A half dozen colleagues and I at the Jet Propulsion Laboratory have been working on the design of an interplanetary extension of the terrestrial Internet. It may take decades to implement this, mission by mission, but to paraphrase: “A man’s reach should exceed his grasp, or what are the heavens for?”

Chapter **[[1]](#footnote-1)** The Network Layer: Control Plane

In this chapter, we’ll complete our journey through the network layer by covering the **control-plane** component of the network layer—the *network-wide* logic that controls not only how a datagram is forwarded among routers along an end-to-end path from the source host to the destination host, but also how network-layer components and services are configured and managed. In **Section 5.2**, we’ll cover traditional routing algorithms for computing least cost paths in a graph; these algorithms are the basis for two widely deployed Internet routing protocols: OSPF and BGP, that we’ll cover in **Sections 5.3** and

5.1 Introduction

Let’s quickly set the context for our study of the network control plane by recalling **Figures 4.2** and **4.3**. There, we saw that the forwarding table (in the case of ­destination-based forwarding) and the flow table (in the case of generalized forwarding) were the principal elements that linked the network layer’s data and control planes. We learned that these tables specify the local data-plane forwarding behavior of a router. We saw that in the case of generalized forwarding, the actions taken (**Section 4.4.2**) could include not only forwarding a packet to a router’s output port, but also dropping a packet, replicating a packet, and/or rewriting layer 2, 3 or 4 packet-header fields.

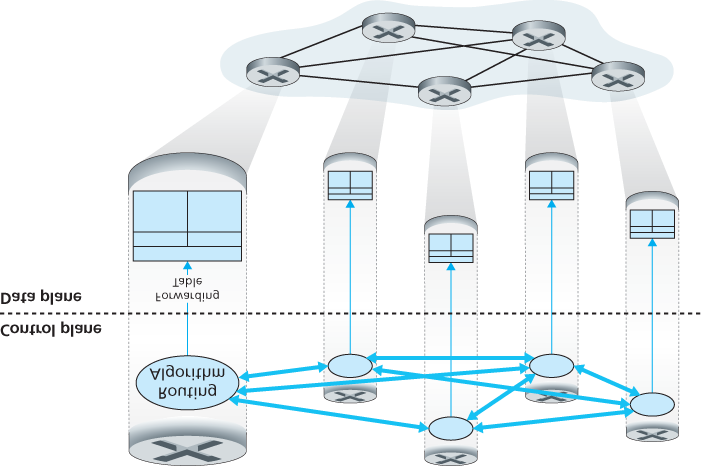
In this chapter, we’ll study how those forwarding and flow tables are computed, maintained and installed. In our introduction to the network layer in **Section 4.1**, we learned that there are two possible approaches for doing so.

**Per-router control.**

**Figure 5.1**

illustrates the case where a routing algorithm runs in each and every

router; both a forwarding and a routing function are contained



## Chapter 5 Review Questions

*SECTION 5.1*

R1. What is meant by a control plane that is based on per-router control? In such cases, when we say the network control and data planes are implemented “monolithically,” what do we mean?

R2. What is meant by a control plane that is based on logically centralized control? In such cases, are the data plane and the control plane implemented within the same device or in separate devices? Explain.

*SECTION 5.2*

R3. Compare and contrast the properties of a centralized and a distributed routing algorithm.

Give an example of a routing protocol that takes a centralized and a decentralized approach.

R4. Compare and contrast link-state and distance-vector routing algorithms.

R5. What is the “count to infinity” problem in distance vector routing?

R6. Is it necessary that every autonomous system use the same intra-AS routing algorithm?

Why or why not?

*SECTIONS 5.3–5.4*

R7. Why are different inter-AS and intra-AS protocols used in the Internet?

R8. True or false: When an OSPF route sends its link state information, it is sent only to those nodes directly attached neighbors. Explain.

R9. What is meant by an *area* in an OSPF autonomous system? Why was the concept of an area introduced?

R10. Define and contrast the following terms: *subnet, prefix*, and *BGP route*.

R11. How does BGP use the NEXT-HOP attribute? How does it use the AS-PATH attribute?

R12. Describe how a network administrator of an upper-tier ISP can implement policy when configuring BGP.

R13. True or false: When a BGP router receives an advertised path from its neighbor, it must add its own identity to the received path and then send that new path on to all of its neighbors.

Explain.

*SECTION 5.5*

R14. Describe the main role of the communication layer, the network-wide state-­management layer, and the network-control application layer in an SDN controller.

R15. Suppose you wanted to implement a new routing protocol in the SDN control plane. At which layer would you implement that protocol? Explain.

R16. What types of messages flow across an SDN controller’s northbound and southbound APIs? Who is the recipient of these messages sent from the controller across the southbound interface, and who sends messages to the controller across the northbound interface?

R17. Describe the purpose of two types of OpenFlow messages (of your choosing) that are sent from a controlled device to the controller. Describe the purpose of two types of Openflow messages (of your choosing) that are send from the controller to a controlled device.

R18. What is the purpose of the service abstraction layer in the OpenDaylight SDN controller?

*SECTIONS 5.6–5.7*

R19. Names four different types of ICMP messages

R20. What two types of ICMP messages are received at the sending host executing the *Traceroute* program?

R21. Define the following terms in the context of SNMP: managing server, ­managed device, network management agent and MIB.

R22. What are the purposes of the SNMP *GetRequest* and *SetRequest* messages?

R23. What is the purpose of the SNMP trap message?

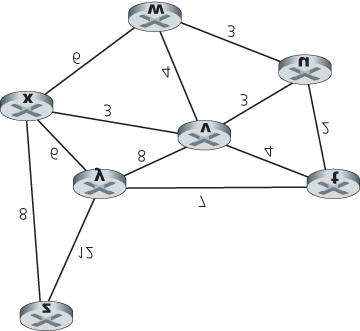
Problems

P1. Looking at **Figure 5.3** , enumerate the paths from *y* to *u* that do not contain any loops.

P2. Repeat Problem P1 for paths from *x* to *z, z* to *u*, and *z* to *w*.

P3. Consider the following network. With the indicated link costs, use Dijkstra’s shortest-path algorithm to compute the shortest path from *x* to all network nodes. Show how the algorithm works by computing a table similar to **Table 5.1** .

|  |
| --- |
| A blue square with a white arrow  AI-generated content may be incorrect.  Dijkstra’s algorithm: discussion and example |



P4. Consider the network shown in Problem P3. Using Dijkstra’s algorithm, and showing your work using a table similar to **Table 5.1** , do the following:

1. Compute the shortest path from *t* to all network nodes.
2. Compute the shortest path from *u* to all network nodes.
3. Compute the shortest path from *v* to all network nodes.
4. Compute the shortest path from *w* to all network nodes.
5. Compute the shortest path from *y* to all network nodes.
6. Compute the shortest path from *z* to all network nodes.

P5. Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node *z*.

A diagram of a graph

AI-generated content may be incorrect.

P6. Consider a general topology (that is, not the specific network shown above) and a synchronous version of the distance-vector algorithm. Suppose that at each iteration, a node exchanges its distance vectors with its neighbors and receives their distance vectors. Assuming that the algorithm begins with each node knowing only the costs to its immediate neighbors, what is the maximum number of iterations required before the distributed algorithm converges?

Justify your answer.

P7. Consider the network fragment shown below. *x* has only two attached neighbors, *w* and *y*. *w* has a minimum-cost path to destination *u* (not shown) of 5, and *y* has a minimum-cost path to *u* of 6. The complete paths from *w* and *y* to *u* (and between *w* and *y*) are not shown. All link costs in the network have strictly positive integer values.

A diagram of a diagram of a diagram

AI-generated content may be incorrect.

1. Give *x*’s distance vector for destinations *w, y,* and *u*.
2. Give a link-cost change for either *c*(*x, w*) or *c*(*x, y*) such that *x* will inform its neighbors of a new minimum-cost path to *u* as a result of executing the distance-vector algorithm.
3. Give a link-cost change for either *c*(*x, w*) or *c*(*x, y*) such that *x* will *not* inform its neighbors of a new minimum-cost path to *u* as a result of executing the distance-vector algorithm.

P8. Consider the three-node topology shown in **Figure 5.6** . Rather than having the link costs shown in **Figure 5.6** , the link costs are c(x,y)=3, c(y,z)=6, c(z,x)=4. Compute the distance tables after the initialization step and after each iteration of a synchronous version of the distancevector algorithm (as we did in our earlier discussion of **Figure 5.6** ).

P9. Consider the count-to-infinity problem in the distance vector routing. Will the count-to-infinity problem occur if we decrease the cost of a link? Why? How about if we connect two nodes which do not have a link?

P10. Argue that for the distance-vector algorithm in **Figure 5.6** , each value in the distance vector *D(x)* is non-increasing and will eventually stabilize in a finite number of steps.

P11. Consider Figure 5.7. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows: c(x,y)=4, c(x,z)=50, c(y,w)=1, c(z,w)=1, c(y,z)=3. Suppose that poisoned reverse is used in the distance-vector routing algorithm.

1. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?
2. Now suppose that the link cost between x and y increases to 60. Will there be a count-toinfinity problem even if poisoned reverse is used? Why or why not? If there is a count-toinfinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.
3. How do you modify *c(y, z)* such that there is no count-to-infinity problem at all if *c(y,x)* changes from 4 to 60?

P12. Describe how loops in paths can be detected in BGP.

P13. Will a BGP router always choose the loop-free route with the shortest ASpath length?

Justify your answer.

P14. Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose

there is *no* physical link between AS2 and AS4.

1. Router 3c learns about prefix *x* from which routing protocol: OSPF, RIP, eBGP, or iBGP?
2. Router 3a learns about *x* from which routing protocol?
3. Router 1c learns about *x* from which routing protocol?
4. Router 1d learns about *x* from which routing protocol?

A diagram of a network

AI-generated content may be incorrect.

P15. Referring to the previous problem, once router 1d learns about *x* it will put an entry (*x*, *I*) in its forwarding table.

1. Will *I* be equal to *I*1 or *I*2 for this entry? Explain why in one sentence.
2. Now suppose that there is a physical link between AS2 and AS4, shown by the dotted line. Suppose router 1d learns that *x* is accessible via AS2 as well as via AS3. Will *I* be set to *I*1 or *I*2? Explain why in one sentence.
3. Now suppose there is another AS, called AS5, which lies on the path between AS2 and AS4 (not shown in diagram). Suppose router 1d learns that *x* is accessible via AS2 AS5 AS4 as well as via AS3 AS4. Will *I* be set to *I*1 or *I*2? Explain why in one sentence.

P16. Consider the following network. ISP B provides national backbone service to regional ISP A. ISP C provides national backbone service to regional ISP D. Each ISP consists of one AS. B and C peer with each other in two places using BGP. Consider traffic going from A to D. B would prefer to hand that traffic over to C on the West Coast (so that C would have to absorb the cost of carrying the traffic cross-country), while C would prefer to get the traffic via its East Coast peering point with B (so that B would have carried the traffic across the country). What BGP mechanism might C use, so that B would hand over A-to-D traffic at its East Coast peering point? To answer this question, you will need to dig into the BGP ­specification.

A map of the united states

AI-generated content may be incorrect.

P17. In Figure 5.13 , consider the path information that reaches stub networks W, X, and Y. Based on the information available at W and X, what are their respective views of the network topology? Justify your answer. The topology view at Y is shown below.

A blue lines with a white circle and a white circle with a black background

AI-generated content may be incorrect.

P18. Consider Figure 5.13 . B would never forward traffic destined to Y via X based on BGP routing. But there are some very popular applications for which data packets go to X first and then flow to Y. Identify one such application, and describe how data packets follow a path not given by BGP routing.

P19. In Figure 5.13 , suppose that there is another stub network V that is a customer of ISP A. Suppose that B and C have a peering relationship, and A is a customer of both B and C. Suppose that A would like to have the traffic destined to W to come from B only, and the traffic destined to V from either B or C. How should A advertise its routes to B and C? What AS routes does C receive?

P20. Suppose ASs X and Z are not directly connected but instead are connected by AS Y. Further suppose that X has a peering agreement with Y, and that Y has a peering agreement with Z. Finally, suppose that Z wants to transit all of Y’s traffic but does not want to transit X’s traffic. Does BGP allow Z to ­implement this policy?

P21. Consider the two ways in which communication occurs between a managing entity and a managed device: request-response mode and trapping. What are the pros and cons of these two approaches, in terms of (1) overhead, (2) notification time when exceptional events occur, and

(3) robustness with respect to lost messages between the managing entity and the device?

P22. In **Section 5.7** we saw that it was preferable to transport SNMP messages in unreliable UDP datagrams. Why do you think the designers of SNMP chose UDP rather than TCP as the transport protocol of choice for SNMP?

## Chapter 6 Review Questions

*SECTIONS 6.1–6.2*

R1. Consider the transportation analogy in **Section 6.1.1** . If the passenger is analagous to a datagram, what is analogous to the link layer frame?

R2. If all the links in the Internet were to provide reliable delivery service, would the TCP reliable delivery service be redundant? Why or why not?

R3. What are some of the possible services that a link-layer protocol can offer to the network layer? Which of these link-layer services have corresponding services in IP? In TCP?

*SECTION 6.3*

R4. Suppose two nodes start to transmit at the same time a packet of length *L* over a broadcast channel of rate *R*. Denote the propagation delay between the two nodes as *d*prop. Will there be a collision if dprop<L/R? Why or why not?

R5. In **Section 6.3** , we listed four desirable characteristics of a broadcast channel. Which of these characteristics does slotted ALOHA have? Which of these characteristics does token passing have?

R6. In CSMA/CD, after the fifth collision, what is the probability that a node chooses K=4? The result K=4 corresponds to a delay of how many ­seconds on a 10 Mbps Ethernet?

R7. Describe polling and token-passing protocols using the analogy of cocktail party interactions.

R8. Why would the token-ring protocol be inefficient if a LAN had a very large perimeter?

*SECTION 6.4*

R9. How big is the MAC address space? The IPv4 address space? The IPv6 address space?

R10. Suppose nodes A, B, and C each attach to the same broadcast LAN (through their adapters). If A sends thousands of IP datagrams to B with each encapsulating frame addressed to the MAC address of B, will C’s adapter process these frames? If so, will C’s adapter pass the IP datagrams in these frames to the network layer C? How would your answers change if A sends frames with the MAC broadcast address?

R11. Why is an ARP query sent within a broadcast frame? Why is an ARP response sent within

a frame with a specific destination MAC address?

R12. For the network in **Figure 6.19** , the router has two ARP modules, each with its own ARP table. Is it possible that the same MAC address appears in both tables?

R13. Compare the frame structures for 10BASE-T, 100BASE-T, and Gigabit ­Ethernet. How do they differ?

R14. Consider **Figure 6.15** . How many subnetworks are there, in the addressing sense of **Section 4.3** ?

R15. What is the maximum number of VLANs that can be configured on a switch supporting the 802.1Q protocol? Why?

R16. Suppose that *N* switches supporting *K* VLAN groups are to be connected via a trunking protocol. How many ports are needed to connect the switches? Justify your answer.

Problems

P1. Suppose the information content of a packet is the bit pattern 1110 0110 1001 1101 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme? Your answer should be such that a minimumlength checksum field is used.

P2. Show (give an example other than the one in **Figure 6.5** ) that two-dimensional parity checks can correct and detect a single bit error. Show (give an example of) a double-bit error that can be detected but not corrected.

P3. Suppose the information portion of a packet (*D* in **Figure 6.3** ) contains 10 bytes consisting of the 8-bit unsigned binary ASCII representation of string “Networking.” Compute the Internet checksum for this data.

P4. Consider the previous problem, but instead suppose these 10 bytes contain

1. the binary representation of the numbers 1 through 10.
2. the ASCII representation of the letters B through K (uppercase).
3. the ASCII representation of the letters b through k (lowercase).

Compute the Internet checksum for this data.

P5. Consider the 5-bit generator, G=10011, and suppose that D has the value 1010101010.

What is the value of R?

P6. Consider the previous problem, but suppose that *D* has the value

1. 1001010101.
2. 0101101010.
3. 1010100000.

P7. In this problem, we explore some of the properties of the CRC. For the ­generator G(=1001) given in **Section 6.2.3** , answer the following questions.

1. Why can it detect any single bit error in data D?
2. Can the above G detect any odd number of bit errors? Why?

P8. In **Section 6.3** , we provided an outline of the derivation of the efficiency of slotted ALOHA.

In this problem we’ll complete the derivation.

1. Recall that when there are *N* active nodes, the efficiency of slotted ALOHA is Np(1−p)N−1. Find the value of *p* that maximizes this expression.
2. Using the value of *p* found in (a), find the efficiency of slotted ALOHA by letting *N* approach infinity. *Hint*: (1−1/N)N approaches 1/*e* as *N* approaches infinity.

P9. Show that the maximum efficiency of pure ALOHA is 1/(2*e*). *Note*: This problem is easy if you have completed the problem above!

P 10. Consider two nodes, A and B, that use the slotted ALOHA protocol to contend for a channel. Suppose node A has more data to transmit than node B, and node A’s retransmission

probability *pA* is greater than node B’s retransmission probability, *pB*.

1. Provide a formula for node A’s average throughput. What is the total efficiency of the protocol with these two nodes?
2. If pA=2pB, is node A’s average throughput twice as large as that of node B? Why or why not? If not, how can you choose *pA* and *pB* to make that happen?
3. In general, suppose there are *N* nodes, among which node A has retransmission probability *2p* and all other nodes have retransmission probability *p.* Provide expressions to compute the average throughputs of node A and of any other node.

P11. Suppose four active nodes—nodes A, B, C and D—are competing for access to a channel using slotted ALOHA. Assume each node has an infinite number of packets to send. Each node attempts to transmit in each slot with probability *p*. The first slot is numbered slot 1, the second slot is numbered slot 2, and so on.

1. What is the probability that node A succeeds for the first time in slot 5?
2. What is the probability that some node (either A, B, C or D) succeeds in slot 4?
3. What is the probability that the first success occurs in slot 3?
4. What is the efficiency of this four-node system?

P12. Graph the efficiency of slotted ALOHA and pure ALOHA as a function of *p* for the following values of *N*:

1. N=15.
2. N=25.
3. N=35.

P13. Consider a broadcast channel with *N* nodes and a transmission rate of *R* bps. Suppose the broadcast channel uses polling (with an additional polling node) for multiple access. Suppose the amount of time from when a node completes transmission until the subsequent node is permitted

to transmit (that is, the polling delay) is *d*poll. Suppose that within a polling round, a given node is allowed to transmit at most *Q* bits. What is the maximum throughput of the broadcast channel?

P14. Consider three LANs interconnected by two routers, as shown in **Figure 6.33** .

1. Assign IP addresses to all of the interfaces. For Subnet 1 use addresses of the form 192.168.1.xxx; for Subnet 2 uses addresses of the form 192.168.2.xxx; and for Subnet 3 use addresses of the form 192.168.3.xxx.
2. Assign MAC addresses to all of the adapters.
3. Consider sending an IP datagram from Host E to Host B. Suppose all of the ARP tables are up to date. Enumerate all the steps, as done for the single-router example in **Section**

**6.4.1** .

1. Repeat (c), now assuming that the ARP table in the sending host is empty (and the other tables are up to date).

P15. Consider **Figure 6.33** . Now we replace the router between subnets 1 and 2 with a switch S1, and label the router between subnets 2 and 3 as R1.

A diagram of a computer network

AI-generated content may be incorrect.

*Figure 6.33 Three subnets, interconnected by routers*

1. Consider sending an IP datagram from Host E to Host F. Will Host E ask router R1 to help forward the datagram? Why? In the Ethernet frame containing the IP datagram, what are the source and destination IP and MAC addresses?
2. Suppose E would like to send an IP datagram to B, and assume that E’s ARP cache does not contain B’s MAC address. Will E perform an ARP query to find B’s MAC address? Why? In the Ethernet frame (containing the IP datagram destined to B) that is delivered to router R1, what are the source and destination IP and MAC addresses?
3. Suppose Host A would like to send an IP datagram to Host B, and neither A’s ARP cache contains B’s MAC address nor does B’s ARP cache contain A’s MAC address. Further suppose that the switch S1’s forwarding table contains entries for Host B and router R1 only. Thus, A will broadcast an ARP request message. What actions will switch S1 perform once it receives the ARP request message? Will router R1 also receive this ARP request message? If so, will R1 forward the message to Subnet 3? Once Host B receives this ARP request message, it will send back to Host A an ARP response message. But will it send an ARP query message to ask for A’s MAC address? Why? What will switch S1 do once it receives an ARP response message from Host B?

P16. Consider the previous problem, but suppose now that the router between subnets 2 and 3 is replaced by a switch. Answer questions (a)–(c) in the previous problem in this new context.

P17. Recall that with the CSMA/CD protocol, the adapter waits K⋅512 bit times after a collision, where *K* is drawn randomly. For K=100, how long does the adapter wait until returning to Step 2 for a 10 Mbps broadcast channel? For a 100 Mbps broadcast channel?

P18. Suppose nodes A and B are on the same 10 Mbps broadcast channel, and the propagation delay between the two nodes is 325 bit times. Suppose CSMA/CD and Ethernet packets are used for this broadcast channel. Suppose node A begins transmitting a frame and, before it finishes, node B begins transmitting a frame. Can A finish transmitting before it detects that B has transmitted? Why or why not? If the answer is yes, then A incorrectly believes that its frame was successfully transmitted without a collision. *Hint*: Suppose at time t=0 bits, A begins transmitting a frame. In the worst case, A transmits a minimum-sized frame of 512+64 bit times. So A would finish transmitting the frame at t=512+64 bit times. Thus, the answer is no, if B’s signal reaches A before bit time t=512+64 bits. In the worst case, when does B’s signal reach A?

P19. Suppose nodes A and B are on the same 10 Mbps broadcast channel, and the propagation delay between the two nodes is 245 bit times. Suppose A and B send Ethernet frames at the same time, the frames collide, and then A and B choose different values of *K* in the CSMA/CD algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide? For our purposes, it suffices to work out the following example. Suppose A and B begin transmission at t=0 bit times. They both detect collisions at t=245 t bit times. Suppose KA=0 and KB=1. At what time does B schedule its retransmission? At what time does A begin transmission? (*Note*: The nodes must wait for an idle channel after returning to Step 2—see protocol.) At what time does A’s signal reach B? Does B refrain from transmitting at its scheduled time?

P20. In this problem, you will derive the efficiency of a CSMA/CD-like multiple access protocol.

In this protocol, time is slotted and all adapters are synchronized to the slots. Unlike slotted ALOHA, however, the length of a slot (in seconds) is much less than a frame time (the time to transmit a frame). Let *S* be the length of a slot. Suppose all frames are of constant length L=kRS, where *R* is the transmission rate of the channel and *k* is a large integer. Suppose there are *N* nodes, each with an infinite number of frames to send. We also assume that dprop<S, so that all nodes can detect a collision before the end of a slot time. The protocol is as follows:

If, for a given slot, no node has possession of the channel, all nodes contend for the channel; in particular, each node transmits in the slot with probability *p*. If exactly one node transmits in the slot, that node takes possession of the channel for the subsequent k−1 slots and transmits its entire frame.

If some node has possession of the channel, all other nodes refrain from transmitting until the node that possesses the channel has finished transmitting its frame. Once this node has transmitted its frame, all nodes contend for the channel.

Note that the channel alternates between two states: the productive state, which lasts exactly *k* slots, and the nonproductive state, which lasts for a random number of slots. Clearly, the channel efficiency is the ratio of k/(k+x), where *x* is the expected number of consecutive unproductive slots.

1. For fixed *N* and *p*, determine the efficiency of this protocol.
2. For fixed *N*, determine the *p* that maximizes the efficiency.
3. Using the *p* (which is a function of *N*) found in (b), determine the efficiency as *N* approaches infinity.
4. Show that this efficiency approaches 1 as the frame length becomes large.

P21. Consider **Figure 6.33** in problem P14. Provide MAC addresses and IP addresses for the interfaces at Host A, both routers, and Host F. Suppose Host A sends a datagram to Host F. Give the source and destination MAC addresses in the frame encapsulating this IP datagram as

the frame is transmitted *(i)* from A to the left router, *(ii)* from the left router to the right router, *(iii)* from the right router to F. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time.

P22. Suppose now that the leftmost router in **Figure 6.33** is replaced by a switch. Hosts A, B, C, and D and the right router are all star-connected into this switch. Give the source and destination

MAC addresses in the frame encapsulating this IP datagram as the frame is transmitted *(i)* from

A to the switch, *(ii)* from the switch to the right router, *(iii)* from the right router to F. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time.

P23. Consider **Figure 6.15** . Suppose that all links are 100 Mbps. What is the maximum total aggregate throughput that can be achieved among the 9 hosts and 2 servers in this network?

You can assume that any host or server can send to any other host or server. Why?

P24. Suppose the three departmental switches in **Figure 6.15** are replaced by hubs. All links are 100 Mbps. Now answer the questions posed in problem P23.

P25. Suppose that *all* the switches in **Figure 6.15** are replaced by hubs. All links are 100 Mbps.

Now answer the questions posed in problem P23.

P26. Let’s consider the operation of a learning switch in the context of a network in which 6 nodes labeled A through F are star connected into an Ethernet switch. Suppose that *(i)* B sends a frame to E, *(ii)* E replies with a frame to B, *(iii)* A sends a frame to B, *(iv)* B replies with a frame to A. The switch table is initially empty. Show the state of the switch table before and after each of these events. For each of these events, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers.

P27. In this problem, we explore the use of small packets for Voice-over-IP applications. One of the drawbacks of a small packet size is that a large fraction of link bandwidth is consumed by

overhead bytes. To this end, suppose that the packet consists of *P* bytes and 5 bytes of header.

1. Consider sending a digitally encoded voice source directly. Suppose the source is encoded at a constant rate of 128 kbps. Assume each packet is entirely filled before the source sends the packet into the network. The time required to fill a packet is the **packetization delay**. In terms of *L*, determine the packetization delay in milliseconds.
2. Packetization delays greater than 20 msec can cause a noticeable and unpleasant echo. Determine the packetization delay for L=1,500 bytes (roughly corresponding to a maximum-sized Ethernet packet) and for L=50 (corresponding to an ATM packet).
3. Calculate the store-and-forward delay at a single switch for a link rate of R=622 Mbps for L=1,500 bytes, and for L=50 bytes.
4. Comment on the advantages of using a small packet size.

P28. Consider the single switch VLAN in **Figure 6.25** , and assume an external router is connected to switch port 1. Assign IP addresses to the EE and CS hosts and router interface. Trace the steps taken at both the network layer and the link layer to transfer an IP datagram from an EE host to a CS host (*Hint:* Reread the discussion of **Figure 6.19** in the text).

P29. Consider the MPLS network shown in **Figure 6.29** , and suppose that routers R5 and R6 are now MPLS enabled. Suppose that we want to perform traffic engineering so that packets from R6 destined for A are switched to A via R6-R4-R3-R1, and packets from R5 destined for A are switched via R5-R4-R2-R1. Show the MPLS tables in R5 and R6, as well as the modified table in R4, that would make this possible.

P30. Consider again the same scenario as in the previous problem, but suppose that packets from R6 destined for D are switched via R6-R4-R3, while packets from R5 destined to D are switched via R4-R2-R1-R3. Show the MPLS tables in all routers that would make this possible.

P31. In this problem, you will put together much of what you have learned about Internet protocols. Suppose you walk into a room, connect to Ethernet, and want to download a Web page. What are all the protocol steps that take place, starting from powering on your PC to getting the Web page? Assume there is nothing in our DNS or browser caches when you power on your PC. (*Hint:* The steps include the use of Ethernet, DHCP, ARP, DNS, TCP, and HTTP protocols.) Explicitly indicate in your steps how you obtain the IP and MAC addresses of a gateway router.

P32. Consider the data center network with hierarchical topology in **Figure 6.30** . Suppose now there are 80 pairs of flows, with ten flows between the first and ninth rack, ten flows between the second and tenth rack, and so on. Further suppose that all links in the network are 10 Gbps, except for the links between hosts and TOR switches, which are 1 Gbps.

1. Each flow has the same data rate; determine the maximum rate of a flow.
2. For the same traffic pattern, determine the maximum rate of a flow for the highly interconnected topology in **Figure 6.31** .
3. Now suppose there is a similar traffic pattern, but involving 20 hosts on each rack and 160 pairs of flows. Determine the maximum flow rates for the two topologies.

P33. Consider the hierarchical network in **Figure 6.30** and suppose that the data center needs to support e-mail and video distribution among other applications. Suppose four racks of servers are reserved for e-mail and four racks are reserved for video. For each of the applications, all four racks must lie below a single tier-2 switch since the tier-2 to tier-1 links do not have sufficient bandwidth to support the intra-application traffic. For the e-mail application, suppose that for 99.9 percent of the time only three racks are used, and that the video application has identical usage patterns.

1. For what fraction of time does the e-mail application need to use a fourth rack? How about for the video application?
2. Assuming e-mail usage and video usage are independent, for what fraction of time do (equivalently, what is the probability that) both applications need their fourth rack?
3. Suppose that it is acceptable for an application to have a shortage of servers for 0.001 percent of time or less (causing rare periods of performance degradation for users).

Discuss how the topology in **Figure 6.31** can be used so that only seven racks are collectively assigned to the two applications (assuming that the topology can support all the traffic).

## Chapter 7 Review Questions

*Section 7.1*

R1. What does it mean for a wireless network to be operating in “infrastructure mode”? If the network is not in infrastructure mode, what mode of operation is it in, and what is the difference between that mode of operation and infrastructure mode?

R2. What are the four types of wireless networks identified in our taxonomy in **Section 7.1** ?

Which of these types of wireless networks have you used?

*Section 7.2*

R3. What are the differences between the following types of wireless channel impairments: path loss, multipath propagation, interference from other sources?

R4. As a mobile node gets farther and farther away from a base station, what are two actions that a base station could take to ensure that the loss probability of a transmitted frame does not increase?

*Sections 7.3 and 7.4*

R5. Describe the role of the beacon frames in 802.11.

R6. True or false: Before an 802.11 station transmits a data frame, it must first send an RTS frame and receive a corresponding CTS frame.

R7. Why are acknowledgments used in 802.11 but not in wired Ethernet?

R8. True or false: Ethernet and 802.11 use the same frame structure.

R9. Describe how the RTS threshold works.

R10. Suppose the IEEE 802.11 RTS and CTS frames were as long as the standard DATA and ACK frames. Would there be any advantage to using the CTS and RTS frames? Why or why not?

R11. **Section 7.3.4** discusses 802.11 mobility, in which a wireless station moves from one BSS to another within the same subnet. When the APs are interconnected with a switch, an AP may need to send a frame with a spoofed MAC address to get the switch to forward the frame properly. Why?

R12. What are the differences between a master device in a Bluetooth network and a base station in an 802.11 network?

R13. What is meant by a super frame in the 802.15.4 Zigbee standard?

R14. What is the role of the “core network” in the 3G cellular data architecture?

R15. What is the role of the RNC in the 3G cellular data network architecture? What role does the RNC play in the cellular voice network?

R16. What is the role of the eNodeB, MME, P-GW, and S-GW in 4G architecture?

R17. What are three important differences between the 3G and 4G cellular ­architectures?

*Sections 7.5 and 7.6*

R18. If a node has a wireless connection to the Internet, does that node have to be mobile? Explain. Suppose that a user with a laptop walks around her house with her laptop, and always accesses the Internet through the same access point. Is this user mobile from a network standpoint? Explain.

R19. What is the difference between a permanent address and a care-of address? Who assigns a care-of address?

R20. Consider a TCP connection going over Mobile IP. True or false: The TCP connection phase between the correspondent and the mobile host goes through the mobile’s home network, but the data transfer phase is directly between the correspondent and the mobile host, bypassing the home network.

*Section 7.7*

R21. What are the purposes of the HLR and VLR in GSM networks? What elements of mobile IP are similar to the HLR and VLR?

R22. What is the role of the anchor MSC in GSM networks?

*Section 7.8*

R23. What are three approaches that can be taken to avoid having a single ­wireless link degrade the performance of an end-to-end transport-layer TCP ­connection?

Problems

P1. Consider the single-sender CDMA example in **Figure 7.5** . What would be the sender’s output (for the 2 data bits shown) if the sender’s CDMA code were (1,−1,1,−1,1,−1,1,−1)?

P2. Consider sender 2 in **Figure 7.6** . What is the sender’s output to the channel (before it is added to the signal from sender 1), Zi,m2?

P3. Suppose that the receiver in **Figure 7.6** wanted to receive the data being sent by sender 2. Show (by calculation) that the receiver is indeed able to recover sender 2’s data from the aggregate channel signal by using sender 2’s code.

P4. For the two-sender, two-receiver example, give an example of two CDMA codes containing 1 and 21 values that do not allow the two receivers to extract the original transmitted bits from the two CDMA senders.

P5. Suppose there are two ISPs providing WiFi access in a particular café, with each ISP operating its own AP and having its own IP address block.

1. Further suppose that by accident, each ISP has configured its AP to operate over channel 11. Will the 802.11 protocol completely break down in this situation? Discuss what happens when two stations, each associated with a different ISP, attempt to transmit at the same time.
2. Now suppose that one AP operates over channel 1 and the other over channel 11. How do your answers change?

P6. In step 4 of the CSMA/CA protocol, a station that successfully transmits a frame begins the CSMA/CA protocol for a second frame at step 2, rather than at step 1. What rationale might the designers of CSMA/CA have had in mind by having such a station not transmit the second frame immediately (if the channel is sensed idle)?

P7. Suppose an 802.11b station is configured to always reserve the channel with the RTS/CTS sequence. Suppose this station suddenly wants to ­transmit 1,000 bytes of data, and all other stations are idle at this time. As a ­function of SIFS and DIFS, and ignoring propagation delay and assuming no bit errors, calculate the time required to transmit the frame and receive the acknowledgment.

P8. Consider the scenario shown in **Figure 7.34** , in which there are four wireless nodes, A, B, C, and D. The radio coverage of the four nodes is shown via the shaded ovals; all nodes share the same frequency. When A transmits, it

A computer screen with a screen and a screen

AI-generated content may be incorrect.

*Figure 7.34 Scenario for problem P8*

can only be heard/received by B; when B transmits, both A and C can hear/receive from B; when C transmits, both B and D can hear/receive from C; when D transmits, only C can hear/receive from D.

Suppose now that each node has an infinite supply of messages that it wants to send to each of the other nodes. If a message’s destination is not an immediate neighbor, then the message must be relayed. For example, if A wants to send to D, a message from A must first be sent to B, which then sends the message to C, which then sends the message to D. Time is slotted, with a message transmission time taking exactly one time slot, e.g., as in slotted Aloha. During a slot, a node can do one of the following: (*i*) send a message, (*ii*) receive a message (if exactly one message is being sent to it), (*iii*) remain silent. As always, if a node hears two or more simultaneous transmissions, a collision occurs and none of the transmitted messages are received successfully. You can assume here that there are no bit-level errors, and thus if exactly one message is sent, it will be received correctly by those within the transmission radius of the sender.

1. Suppose now that an omniscient controller (i.e., a controller that knows the state of every node in the network) can command each node to do whatever it (the omniscient controller) wishes, i.e., to send a message, to receive a message, or to remain silent. Given this omniscient controller, what is the maximum rate at which a data message can be transferred from C to A, given that there are no other messages between any other source/destination pairs?
2. Suppose now that A sends messages to B, and D sends messages to C. What is the combined maximum rate at which data messages can flow from A to B and from D to C?
3. Suppose now that A sends messages to B, and C sends messages to D. What is the combined maximum rate at which data messages can flow from A to B and from C to D?
4. Suppose now that the wireless links are replaced by wired links. Repeat questions (a) through (c) again in this wired scenario.
5. Now suppose we are again in the wireless scenario, and that for every data message sent from source to destination, the destination will send an ACK message back to the source (e.g., as in TCP). Also suppose that each ACK message takes up one slot.

Repeat questions (a)–(c) above for this scenario.

P9. Describe the format of the 802.15.1 Bluetooth frame. You will have to do some reading outside of the text to find this information. Is there anything in the frame format that inherently limits the number of active nodes in an 802.15.1 network to eight active nodes? Explain.

P10. Consider the following idealized LTE scenario. The downstream channel (see **Figure 7.21** ) is slotted in time, across F frequencies. There are four nodes, A, B, C, and D, reachable from the base station at rates of 10 Mbps, 5 Mbps, 2.5 Mbps, and 1 Mbps, respectively, on the downstream channel. These rates assume that the base station utilizes all time slots available on all F frequencies to send to just one station. The base station has an infinite amount of data to send to each of the nodes, and can send to any one of these four nodes using any of the F frequencies during any time slot in the ­downstream sub-frame.

1. What is the maximum rate at which the base station can send to the nodes, assuming it

can send to any node it chooses during each time slot? Is your solution fair? Explain and define what you mean by “fair.”

1. If there is a fairness requirement that each node must receive an equal amount of data during each one second interval, what is the average transmission rate by the base station (to all nodes) during the downstream sub-frame? Explain how you arrived at your answer.
2. Suppose that the fairness criterion is that any node can receive at most twice as much data as any other node during the sub-frame. What is the average transmission rate by the base station (to all nodes) during the sub-frame? Explain how you arrived at your answer.

P11. In **Section 7.5** , one proposed solution that allowed mobile users to maintain their IP addresses as they moved among foreign networks was to have a foreign network advertise a highly specific route to the mobile user and use the existing routing infrastructure to propagate this information throughout the network. We identified scalability as one concern. Suppose that when a mobile user moves from one network to another, the new foreign network advertises a specific route to the mobile user, and the old foreign network withdraws its route. Consider how routing information propagates in a distance-vector algorithm (particularly for the case of interdomain routing among networks that span the globe).

1. Will other routers be able to route datagrams immediately to the new foreign network as soon as the foreign network begins advertising its route?
2. Is it possible for different routers to believe that different foreign networks contain the mobile user?
3. Discuss the timescale over which other routers in the network will eventually learn the path to the mobile users.

P12. Suppose the correspondent in **Figure 7.23** were mobile. Sketch the additional networklayer infrastructure that would be needed to route the datagram from the original mobile user to the (now mobile) correspondent. Show the structure of the datagram(s) between the original mobile user and the (now mobile) correspondent, as in **Figure 7.24** .

P13. In mobile IP, what effect will mobility have on end-to-end delays of datagrams between the source and destination?

P14. Consider the chaining example discussed at the end of **Section 7.7.2** . Suppose a mobile user visits foreign networks A, B, and C, and that a correspondent begins a connection to the mobile user when it is resident in foreign ­network A. List the sequence of messages between foreign agents, and between foreign agents and the home agent as the mobile user moves from network A to network B to network C. Next, suppose chaining is not performed, and the correspondent (as well as the home agent) must be explicitly notified of the changes in the mobile user’s care-of address. List the sequence of messages that would need to be exchanged in this second scenario.

P15. Consider two mobile nodes in a foreign network having a foreign agent. Is it possible for the two mobile nodes to use the same care-of address in mobile IP? Explain your answer.

P16. In our discussion of how the VLR updated the HLR with information about the mobile’s current location, what are the advantages and disadvantages of providing the MSRN as opposed to the address of the VLR to the HLR?

## Chapter 8 Review Problems

*SECTION 8.1*

R1. What are the differences between message confidentiality and message integrity? Can you have confidentiality without integrity? Can you have integrity without confidentiality? Justify your answer.

R2. Internet entities (routers, switches, DNS servers, Web servers, user end systems, and so on) often need to communicate securely. Give three specific example pairs of Internet entities that may want secure communication.

*SECTION 8.2*

R3. From a service perspective, what is an important difference between a symmetric-key system and a public-key system?

R4. Suppose that an intruder has an encrypted message as well as the decrypted version of that message. Can the intruder mount a ciphertext-only attack, a known-plaintext attack, or a chosenplaintext attack?

R5. Consider an 8-block cipher. How many possible input blocks does this cipher have? How many possible mappings are there? If we view each mapping as a key, then how many possible keys does this cipher have?

R6. Suppose *N* people want to communicate with each of N−1 other people using symmetric key encryption. All communication between any two people, *i* and *j*, is visible to all other people in this group of *N*, and no other person in this group should be able to decode their communication. How many keys are required in the system as a whole? Now suppose that public key encryption is used. How many keys are required in this case?

R7. Suppose n=10,000, a=10,023, and b=10,004. Use an identity of modular arithmetic to calculate in your head (a⋅b)mod n.

R8. Suppose you want to encrypt the message 10101111 by encrypting the decimal number that corresponds to the message. What is the decimal number?

*SECTIONS 8.3–8.4*

R9. In what way does a hash provide a better message integrity check than a checksum (such as the Internet checksum)?

R10. Can you “decrypt” a hash of a message to get the original message? Explain your answer.

R11. Consider a variation of the MAC algorithm (**Figure 8.9** ) where the sender sends

(m, H(m)+s), where H(m)+s is the concatenation of *H*(*m*) and *s*. Is this variation flawed? Why or why not?

R12. What does it mean for a signed document to be verifiable and nonforgeable?

R13. In what way does the public-key encrypted message hash provide a better digital signature than the public-key encrypted message?

R14. Suppose [**certifier.com**](http://certifier.com/) creates a certificate for [**foo.com**](http://foo.com/). Typically, the entire certificate would be encrypted with [**certifier.com**’](http://certifier.com/)s public key. True or false?

R15. Suppose Alice has a message that she is ready to send to anyone who asks. Thousands of people want to obtain Alice’s message, but each wants to be sure of the integrity of the message. In this context, do you think a MAC-based or a digital-signature-based integrity scheme is more suitable? Why?

R16. What is the purpose of a nonce in an end-point authentication protocol?

R17. What does it mean to say that a nonce is a once-in-a-lifetime value? In whose lifetime?

R18. Is the message integrity scheme based on HMAC susceptible to playback attacks? If so, how can a nonce be incorporated into the scheme to remove this susceptibility?

*SECTIONS 8.5–8.8*

R19. Suppose that Bob receives a PGP message from Alice. How does Bob know for sure that Alice created the message (rather than, say, Trudy)? Does PGP use a MAC for message integrity?

R20. In the SSL record, there is a field for SSL sequence numbers. True or false?

R21. What is the purpose of the random nonces in the SSL handshake?

R22. Suppose an SSL session employs a block cipher with CBC. True or false: The server sends to the client the IV in the clear.

R23. Suppose Bob initiates a TCP connection to Trudy who is pretending to be Alice. During the handshake, Trudy sends Bob Alice’s certificate. In what step of the SSL handshake algorithm will Bob discover that he is not communicating with Alice?

R24. Consider sending a stream of packets from Host A to Host B using IPsec. Typically, a new SA will be established for each packet sent in the stream. True or false?

R25. Suppose that TCP is being run over IPsec between headquarters and the branch office in

**Figure 8.28** . If TCP retransmits the same packet, then the two corresponding packets sent by R1 packets will have the same sequence number in the ESP header. True or false?

R26. An IKE SA and an IPsec SA are the same thing. True or false?

R27. Consider WEP for 802.11. Suppose that the data is 10101100 and the keystream is 1111000. What is the resulting ciphertext?

R28. In WEP, an IV is sent in the clear in every frame. True or false?

*SECTION 8.9*

R29. Stateful packet filters maintain two data structures. Name them and briefly describe what they do.

R30. Consider a traditional (stateless) packet filter. This packet filter may filter packets based on TCP flag bits as well as other header fields. True or false?

R31. In a traditional packet filter, each interface can have its own access control list. True or false?

R32. Why must an application gateway work in conjunction with a router filter to be effective?

R33. Signature-based IDSs and IPSs inspect into the payloads of TCP and UDP segments. True or false?

Problems

P1. Using the monoalphabetic cipher in **Figure 8.3** , encode the message “This is an easy problem.” Decode the message “rmij’u uamu xyj.”

P2. Show that Trudy’s known-plaintext attack, in which she knows the (ciphertext, plaintext) translation pairs for seven letters, reduces the number of possible substitutions to be checked in the example in **Section 8.2.1** by approximately 109.

P3. Consider the polyalphabetic system shown in **Figure 8.4** . Will a chosen-plaintext attack that is able to get the plaintext encoding of the message “The quick brown fox jumps over the lazy dog.” be sufficient to decode all messages? Why or why not?

P4. Consider the block cipher in **Figure 8.5** . Suppose that each block cipher *Ti* simply reverses the order of the eight input bits (so that, for example, 11110000 becomes 00001111). Further suppose that the 64-bit scrambler does not modify any bits (so that the output value of the *m*th bit is equal to the input value of the *m*th bit). (a) With n=3 and the original 64-bit input equal to 10100000 repeated eight times, what is the value of the output? (b) Repeat part (a) but now change the last bit of the original 64-bit input from a 0 to a 1. (c) Repeat parts (a) and (b) but now suppose that the 64-bit scrambler inverses the order of the 64 bits.

P5. Consider the block cipher in **Figure 8.5** . For a given “key” Alice and Bob would need to keep eight tables, each 8 bits by 8 bits. For Alice (or Bob) to store all eight tables, how many bits of storage are necessary? How does this number compare with the number of bits required for a full-table 64-bit block cipher?

P6. Consider the 3-bit block cipher in **Table 8.1** . Suppose the plaintext is 100100100. (a) Initially assume that CBC is not used. What is the resulting ciphertext? (b) Suppose Trudy sniffs the ciphertext. Assuming she knows that a 3-bit block cipher without CBC is being employed (but doesn’t know the specific cipher), what can she surmise? (c) Now suppose that CBC is used

with IV=111. What is the resulting ciphertext?

P7. (a) Using RSA, choose p=3 and q=11, and encode the word “dog” by encrypting each letter separately. Apply the decryption algorithm to the encrypted version to recover the original plaintext message. (b) Repeat part (a) but now encrypt “dog” as one message *m*.

P8. Consider RSA with p=5 and q=11.

1. What are *n* and *z*?
2. Let *e* be 3. Why is this an acceptable choice for *e*?
3. Find *d* such that de=1 (mod *z*) and d<160.
4. Encrypt the message m=8 using the key (*n*, *e*). Let *c* denote the corresponding ciphertext. Show all work. *Hint:* To simplify the calculations, use the fact:

[ (a mod n)⋅(b mod n)]mod n=(a⋅b)modn

P9. In this problem, we explore the Diffie-Hellman (DH) public-key encryption algorithm, which allows two entities to agree on a shared key. The DH algorithm makes use of a large prime number *p* and another large number *g* less than *p*. Both *p* and *g* are made public (so that an attacker would know them). In DH, Alice and Bob each independently choose secret keys, *SA* and *SB*, respectively. Alice then computes her public key, *TA*, by raising *g* to *SA* and then taking mod *p*. Bob similarly computes his own public key *TB* by raising *g* to *SB* and then taking mod *p*. Alice and Bob then exchange their public keys over the Internet. Alice then calculates the shared secret key *S* by raising *TB* to *SA* and then taking mod *p*. Similarly, Bob calculates the shared key *S*′ by raising *TA* to *SB* and then taking mod *p*.

1. Prove that, in general, Alice and Bob obtain the same symmetric key, that is, prove S=S′.
2. With *p* = 11 and *g* = 2, suppose Alice and Bob choose private keys SA=5 and SB=12, respectively. Calculate Alice’s and Bob’s public keys, *TA* and *TB*. Show all work.
3. Following up on part (b), now calculate *S* as the shared symmetric key. Show all work.
4. Provide a timing diagram that shows how Diffie-Hellman can be attacked by a man-inthe-middle. The timing diagram should have three vertical lines, one for Alice, one for Bob, and one for the attacker Trudy.

P10. Suppose Alice wants to communicate with Bob using symmetric key cryptography using a session key *KS*. In **Section 8.2** , we learned how public-key cryptography can be used to distribute the session key from Alice to Bob. In this problem, we explore how the session key can be distributed—without public key cryptography—using a key distribution center (KDC). The KDC is a server that shares a unique secret symmetric key with each registered user. For Alice and Bob, denote these keys by *KA-KDC* and *KB-KDC*. Design a scheme that uses the KDC to distribute *KS* to Alice and Bob. Your scheme should use three messages to distribute the session key: a message from Alice to the KDC; a message from the KDC to Alice; and finally a message from Alice to Bob. The first message is *KA-KDC* (*A*, *B*). Using the notation, *KA-KDC*, *KB-KDC*, *S, A, and B* answer the following questions.

1. What is the second message?
2. What is the third message?

P11. Compute a third message, different from the two messages in **Figure 8.8** , that has the same checksum as the messages in **Figure 8.8** .

P12. Suppose Alice and Bob share two secret keys: an authentication key *S1* and a symmetric encryption key *S2*. Augment **Figure 8.9** so that both integrity and confidentiality are provided.

P13. In the BitTorrent P2P file distribution protocol (see **Chapter 2** ), the seed breaks the file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a small subset of peers in the torrent. These unsuspecting peers then redistribute the bogus blocks to other peers, which in turn redistribute the bogus blocks to even more peers. Thus, it is critical for BitTorrent to have a mechanism that allows a peer to verify the integrity of a block, so that it doesn’t redistribute bogus blocks. Assume that when a peer joins a torrent, it initially gets a .torrent file from a *fully* trusted source. Describe a simple scheme that allows peers to verify the integrity of blocks.

P14. The OSPF routing protocol uses a MAC rather than digital signatures to provide message integrity. Why do you think a MAC was chosen over digital signatures?

P15. Consider our authentication protocol in **Figure 8.18** in which Alice authenticates herself to Bob, which we saw works well (i.e., we found no flaws in it). Now suppose that while Alice is authenticating herself to Bob, Bob must authenticate himself to Alice. Give a scenario by which

Trudy, pretending to be Alice, can now authenticate herself to Bob as Alice. (*Hint*: Consider that the sequence of operations of the protocol, one with Trudy initiating and one with Bob initiating, can be arbitrarily interleaved. Pay particular attention to the fact that both Bob and Alice will use a nonce, and that if care is not taken, the same nonce can be used maliciously.)

P16. A natural question is whether we can use a nonce and public key cryptography to solve the end-point authentication problem in **Section 8.4** . Consider the following natural protocol: (1)

Alice sends the message “*I am Alice*” to Bob. (2) Bob chooses a nonce, *R*, and sends it to

Alice. (3) Alice uses her *private* key to encrypt the nonce and sends the resulting value to Bob.

(4) Bob applies Alice’s public key to the received message. Thus, Bob computes *R* and authenticates Alice.

1. Diagram this protocol, using the notation for public and private keys employed in the textbook.
2. Suppose that certificates are not used. Describe how Trudy can become a “woman-inthe-middle” by intercepting Alice’s messages and then ­pretending to be Alice to Bob.

P17. **Figure 8.19** shows the operations that Alice must perform with PGP to provide confidentiality, authentication, and integrity. Diagram the corresponding operations that Bob must perform on the package received from Alice.

P18. Suppose Alice wants to send an e-mail to Bob. Bob has a public-private key pair (KB+,KB−), and Alice has Bob’s certificate. But Alice does not have a public, private key pair.

Alice and Bob (and the entire world) share the same hash function H(⋅).

1. In this situation, is it possible to design a scheme so that Bob can verify that Alice created the message? If so, show how with a block diagram for Alice and Bob.
2. Is it possible to design a scheme that provides confidentiality for sending the message from Alice to Bob? If so, show how with a block diagram for Alice and Bob.

P19. Consider the Wireshark output below for a portion of an SSL session.

1. Is Wireshark packet 112 sent by the client or server?
2. What is the server’s IP address and port number?
3. Assuming no loss and no retransmissions, what will be the sequence number of the next TCP segment sent by the client?
4. How many SSL records does Wireshark packet 112 contain?
5. Does packet 112 contain a Master Secret or an Encrypted Master Secret or neither?
6. Assuming that the handshake type field is 1 byte and each length field is 3 bytes, what are the values of the first and last bytes of the Master Secret (or Encrypted Master Secret)?
7. The client encrypted handshake message takes into account how many SSL records?
8. The server encrypted handshake message takes into account how many SSL records?

P20. In **Section 8.6.1** , it is shown that without sequence numbers, Trudy (a woman-in-the middle) can wreak havoc in an SSL session by interchanging TCP segments. Can Trudy do something similar by deleting a TCP segment? What does she need to do to succeed at the deletion attack? What effect will it have?

A screenshot of a computer

AI-generated content may be incorrect.

(Wireshark screenshot reprinted by permission of the Wireshark Foundation.)

P21. Suppose Alice and Bob are communicating over an SSL session. Suppose an attacker, who does not have any of the shared keys, inserts a bogus TCP segment into a packet stream with correct TCP checksum and sequence numbers (and correct IP addresses and port numbers). Will SSL at the receiving side accept the bogus packet and pass the payload to the receiving application? Why or why not?

P22. The following true/false questions pertain to **Figure 8.28** .

1. When a host in 172.16.1/24 sends a datagram to an [**Amazon.com**](http://amazon.com/) server, the router R1 will encrypt the datagram using IPsec.
2. When a host in 172.16.1/24 sends a datagram to a host in 172.16.2/24, the router R1 will change the source and destination address of the IP datagram.
3. Suppose a host in 172.16.1/24 initiates a TCP connection to a Web server in 172.16.2/24. As part of this connection, all datagrams sent by R1 will have protocol number 50 in the left-most IPv4 header field.
4. Consider sending a TCP segment from a host in 172.16.1/24 to a host in 172.16.2/24. Suppose the acknowledgment for this segment gets lost, so that TCP resends the segment. Because IPsec uses sequence numbers, R1 will not resend the TCP segment.

P23. Consider the example in **Figure 8.28** . Suppose Trudy is a woman-in-the-middle, who can insert datagrams into the stream of datagrams going from R1 and R2. As part of a replay attack, Trudy sends a duplicate copy of one of the datagrams sent from R1 to R2. Will R2 decrypt the duplicate datagram and forward it into the branch-office network? If not, describe in detail how R2 detects the duplicate datagram.

P24. Consider the following pseudo-WEP protocol. The key is 4 bits and the IV is 2 bits. The IV is appended to the end of the key when generating the keystream. Suppose that the shared secret key is 1010. The keystreams for the four possible inputs are as follows:

101000: 0010101101010101001011010100100 . . .

101001: 1010011011001010110100100101101 . . .

101010: 0001101000111100010100101001111 . . .

101011: 1111101010000000101010100010111 . . .

Suppose all messages are 8 bits long. Suppose the ICV (integrity check) is 4 bits long, and is calculated by XOR-ing the first 4 bits of data with the last 4 bits of data. Suppose the pseudoWEP packet consists of three fields: first the IV field, then the message field, and last the ICV field, with some of these fields encrypted.

1. We want to send the message m=10100000 using the IV=11 and using WEP. What will be the values in the three WEP fields?
2. Show that when the receiver decrypts the WEP packet, it recovers the message and the ICV.
3. Suppose Trudy intercepts a WEP packet (not necessarily with the IV=11) and wants to modify it before forwarding it to the receiver. Suppose Trudy flips the first ICV bit. Assuming that Trudy does not know the keystreams for any of the IVs, what other bit(s) must Trudy also flip so that the received packet passes the ICV check?
4. Justify your answer by modifying the bits in the WEP packet in part (a), decrypting the resulting packet, and verifying the integrity check.

P25. Provide a filter table and a connection table for a stateful firewall that is as restrictive as possible but accomplishes the following:

1. Allows all internal users to establish Telnet sessions with external hosts.
2. Allows external users to surf the company Web site at 222.22.0.12.
3. But otherwise blocks all inbound and outbound traffic.

The internal network is 222.22/16. In your solution, suppose that the connection table is currently caching three connections, all from inside to outside. You’ll need to invent appropriate IP addresses and port numbers.

P26. Suppose Alice wants to visit the Web site [**activist.com**](http://activist.com/) using a TOR-like ­service. This service uses two non-colluding proxy servers, Proxy1 and Proxy2. Alice first obtains the certificates (each containing a public key) for Proxy1 and Proxy2 from some central server. Denote K1+(),K2+(),K1−(), and K2−() for the encryption/decryption with public and private RSA keys.

* 1. Using a timing diagram, provide a protocol (as simple as possible) that enables Alice to establish a shared session key *S*1 with Proxy1. Denote *S*1(*m*) for encryption/decryption of data *m* with the shared key *S*1.
  2. Using a timing diagram, provide a protocol (as simple as possible) that allows Alice to establish a shared session key *S*2 with Proxy2 *without revealing her IP address to*

*Proxy2*.

* 1. Assume now that shared keys *S*1 and *S*2 are now established. Using a timing diagram, provide a protocol (as simple as possible and *not using public-key cryptography*) that allows Alice to request an html page from [**activist.com**](http://activist.com/) *without revealing her IP address to Proxy2* and *without revealing to Proxy1 which site she is visiting*. Your diagram should end with an HTTP request arriving at [**activist.com**](http://activist.com/).

1. **.4**, respectively. As we’ll see, OSPF is a routing protocol that operates within a single ISP’s network. BGP is a routing protocol that serves to interconnect all of the networks in the Internet; BGP is thus often referred to as the “glue” that holds the Internet together. Traditionally, control-plane routing protocols have been implemented together with data-plane forwarding functions, monolithically, within a router. As we learned in the introduction to **Chapter 4**, software-defined networking (SDN) makes a clear separation between the data and control planes, implementing control-plane functions in a separate “controller” service that is distinct, and remote, from the forwarding components of the routers it controls. We’ll cover SDN controllers in **Section 5.5**.

   In **Sections 5.6** and **5.7** we’ll cover some of the nuts and bolts of managing an IP network: ICMP (the Internet Control Message Protocol) and SNMP (the Simple Network Management Protocol). [↑](#footnote-ref-1)