studocu

Computer networking A top-down approach (Solutions to review questions and problems) by J.F. Kurose, K.W. Ross (z-lib

Financial Accounting (Gachon University)

Scan to open on Studocu

Studocu is not sponsored or endorsed by any college or university
Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Computer Networking: A Top-Down Approach Featuring the Internet, 4th Edition

Solutions to Review Questions and Problems

Version Date: June 21, 2007

This document contains the solutions to review questions and problems for the 4th edition of *Computer Networking: A Top-Down Approach Featuring the Internet* by Jim Kurose and Keith Ross. These solutions are being made available to instructors ONLY. Please do NOT copy or distribute this document to others (even other instructors). Please do not post any solutions on a publicly-available Web site. We'll be happy to provide a copy (up-to-date) of this solution manual ourselves to anyone who asks.

All material © copyright 1996-2007 by J.F. Kurose and K.W. Ross. All rights reserved

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 1 Review Questions

1. There is no difference. Throughout this text, the words "host" and "end system" are used interchangeably. End systems include PCs, workstations, Web servers, mail servers, Internet-connected PDAs, WebTVs, etc.

2. Suppose Alice, an ambassador of country A wants to invite Bob, an ambassador of country B, over for dinner. Alice doesn't simply just call Bob on the phone and say, "come to our dinner table now". Instead, she calls Bob and suggests a date and time. Bob may respond by saying he's not available that particular date, but he is available another date. Alice and Bob continue to send "messages" back and forth until they agree on a date and time. Bob then shows up at the embassy on the agreed date, hopefully not more than 15 minutes before or after the agreed time. Diplomatic protocols also allow for either Alice or Bob to politely cancel the engagement if they have reasonable excuses.

3. A networking program usually has two programs, each running on a different host, communicating with each other. The program that initiates the communication is the client. Typically, the client program requests and receives services from the server program.

4. 1. Dial-up modem over telephone line: residential; 2. DSL over telephone line: residential or small office; 3. Cable to HFC: residential; 4. 100 Mbps switched Etherent: company; 5. Wireless LAN: mobile; 6. Cellular mobile access (for example, WAP): mobile

5. HFC bandwidth is shared among the users. On the downstream channel, all packets emanate from a single source, namely, the head end. Thus, there are no collisions in the downstream channel.

6. Current possibilities include: dial-up; DSL; cable modem; fiber-to-the-home.

7. Ethernet LANs have transmission rates of 10 Mbps, 100 Mbps, 1 Gbps and 10 Gbps. For an X Mbps Ethernet (where X = 10, 100, 1,000 or 10,000), a user can continuously transmit at the rate X Mbps if that user is the only person sending data. If there are more than one active user, then each user cannot continuously transmit at X Mbps.

8. Ethernet most commonly runs over twisted-pair copper wire and "thin" coaxial cable. It also can run over fibers optic links and thick coaxial cable.

9. Dial up modems: up to 56 Kbps, bandwidth is dedicated; ISDN: up to 128 kbps, bandwidth is dedicated; ADSL: downstream channel is .5-8 Mbps, upstream channel

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

is up to 1 Mbps, bandwidth is dedicated; HFC, downstream channel is 10-30 Mbps and upstream channel is usually less than a few Mbps, bandwidth is shared.

10. There are two most popular wireless Internet access technologies today:
    a) Wireless LAN
       In a wireless LAN, wireless users transmit/receive packets to/from a base station (wireless access point) within a radius of few tens of meters. The base station is typically connected to the wired Internet and thus serves to connect wireless users to the wired network.

    b) Wide-area wireless access network
       In these systems, packets are transmitted over the same wireless infrastructure used for cellular telephony, with the base station thus being managed by a telecommunications provider. This provides wireless access to users within a radius of tens of kilometers of the base station.

11. A circuit-switched network can guarantee a certain amount of end-to-end bandwidth for the duration of a call. Most packet-switched networks today (including the Internet) cannot make any end-to-end guarantees for bandwidth.

12. In a packet switched network, the packets from different sources flowing on a link do not follow any fixed, pre-defined pattern. In TDM circuit switching, each host gets the same slot in a revolving TDM frame.

13. At time t_0 the sending host begins to transmit. At time $t_1 = L/R_1$, the sending host completes transmission and the entire packet is received at the router (no propagation delay). Because the router has the entire packet at time $t1$, it can begin to transmit the packet to the receiving host at time t_1. At time $t_2 = t_1 + L/R_2$, the router completes transmission and the entire packet is received at the receiving host (again, no propagation delay). Thus, the end-to-end delay is $L/R_1 + L/R_2$.

14. A tier-1 ISP connects to all other tier-1 ISPs; a tier-2 ISP connects to only a few of the tier-1 ISPs. Also, a tier-2 ISP is a customer of one or more tier-1.

15. a) 2 users can be supported because each user requires half of the link bandwidth.

    b) Since each user requires 1Mbps when transmitting, if two or fewer users transmit simultaneously, a maximum of 2Mbps will be required. Since the available bandwidth of the shared link is 2Mbps, there will be no queuing delay before the link. Whereas, if three users transmit simultaneously, the bandwidth required will be 3Mbps which is more than the available bandwidth of the shared link. In this case, there will be queuing delay before the link.

    c) Probability that a given user is transmitting = 0.2

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

d) Probability that all three users are transmitting simultaneously = $\binom{3}{3} p^3 (1-p)^{3-3}$ $= (0.2)^3 = 0.008$. Since the queue grows when all the users are transmitting, the fraction of time during which the queue grows (which is equal to the probability that all three users are transmitting simultaneously) is 0.008.

16. The delay components are processing delays, transmission delays, propagation delays, and queuing delays. All of these delays are fixed, except for the queuing delays, which are variable.

17. Java Applet

18. 10msec; d/s; no; no

19. a) 500 kbps
    b) 62.5 seconds
    c) 100kbps; 312.5 seconds

20. End system A breaks the large file into chunks. To each chunk, it adds header generating multiple packets from the file. The header in each packet includes the address of the destination: end system B. The packet switch uses the destination address to determine the outgoing link. Asking which road to take is analogous to a packet asking which outgoing link it should be forwarded on, given the packet's address.

21. Java Applet

22. Five generic tasks are error control, flow control, segmentation and reassembly, multiplexing, and connection setup. Yes, these tasks can be duplicated at different layers. For example, error control is often provided at more than one layer.

23. The five layers in the Internet protocol stack are – from top to bottom – the application layer, the transport layer, the network layer, the link layer, and the physical layer. The principal responsibilities are outlined in Section 1.5.1.

24. Application-layer message: data which an application wants to send and passed onto the transport layer; transport-layer segment: generated by the transport layer and encapsulates application-layer message with transport layer header; network-layer datagram: encapsulates transport-layer segment with a network-layer header; link-layer frame: encapsulates network-layer datagram with a link-layer header.

25. Routers process layers 1 through 3. (This is a little bit of a white lie, as modern routers sometimes act as firewalls or caching components, and process layer four as well.) Link layer switches process layers 1 through 2. Hosts process all five layers.

26. a) Virus

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Requires some form of human interaction to spread. Classic example: E-mail viruses.

b)Worms

No user replication needed. Worm in infected host scans IP addresses and port numbers, looking for vulnerable processes to infect.

c) Trojan horse

Hidden, devious part of some otherwise useful software.

27. Creation of a botnet requires an attacker to find vulnerability in some application or system (e.g. exploiting the buffer overflow vulnerability that might exist in an application). After finding the vulnerability, the attacker needs to scan for hosts that are vulnerable. The target is basically to compromise a series of systems by exploiting that particular vulnerability. Any system that is part of the botnet can automatically scan its environment and propagate by exploiting the vulnerability. An important property of such botnets is that the originator of the botnet can remotely control and issue commands to all the nodes in the botnet. Hence, it becomes possible for the attacker to issue a command to all the nodes, that target a single node (for example, all nodes in the botnet might be commanded by the attacker to send a TCP SYN message to the target, which might result in a TCP SYN flood attack at the target).

28. Trudy can pretend to be Bob to Alice (and vice-versa) and partially or completely modify the message(s) being sent from Bob to Alice. For example, she can easily change the phrase "Alice, I owe you $1000" to "Alice, I owe you $10,000". Furthermore, Trudy can even drop the packets that are being sent by Bob to Alice (and vise-versa), even if the packets from Bob to Alice are encrypted.

Chapter 1 Problems:

Problem 1.

There is no single right answer to this question. Many protocols would do the trick. Here's a simple answer below:

Messages from ATM machine to Server

```
Msg name             purpose
--------             -------
HELO <userid>        Let server know that there is a card in the
                     ATM machine
                     ATM card transmits user ID to Server
PASSWD <passwd>      User enters PIN, which is sent to server
BALANCE              User requests balance
WITHDRAWL <amount>   User asks to withdraw money
BYE                  user all done
```

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Messages from Server to ATM machine (display)

```
Msg name            purpose
--------            -------
PASSWD              Ask user for PIN (password)
OK                  last requested operation (PASSWD, WITHDRAWL)
                    OK
ERR                 last requested operation (PASSWD, WITHDRAWL)
                    in ERROR
AMOUNT <amt>        sent in response to BALANCE request
BYE                 user done, display welcome screen at ATM
```

Correct operation:

```
client                          server

HELO (userid)   --------------> (check if valid userid)
                <-------------  PASSWD
PASSWD <passwd> --------------> (check password)
                <-------------  OK (password is OK)
BALANCE         -------------->
                <-------------  AMOUNT <amt>
WITHDRAWL <amt> --------------> check if enough $ to cover
                                withdrawl
                <-------------  OK
ATM dispenses $
BYE             -------------->
                <-------------  BYE
```

In situation when there's not enough money:

```
HELO (userid)   --------------> (check if valid userid)
                <-------------  PASSWD
PASSWD <passwd> --------------> (check password)
                <-------------  OK (password is OK)
BALANCE         -------------->
                <-------------  AMOUNT <amt>
WITHDRAWL <amt> --------------> check  if  enough  $  to  cover
withdrawl
                <-------------  ERR (not enough funds)
error msg displayed
no $ given out
BYE             -------------->
                <-------------  BYE
```

Problem 2.

a) A circuit-switched network would be well suited to the application described, because the application involves long sessions with predictable smooth bandwidth requirements. Since the transmission rate is known and not bursty, bandwidth can be reserved for each application session circuit with no significant waste. In addition, we

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

need not worry greatly about the overhead costs of setting up and tearing down a circuit connection, which are amortized over the lengthy duration of a typical application session.

b) Given such generous link capacities, the network needs no congestion control mechanism. In the worst (most potentially congested) case, all the applications simultaneously transmit over one or more particular network links. However, since each link offers sufficient bandwidth to handle the sum of all of the applications' data rates, no congestion (very little queuing) will occur.

Problem 3.

a) We can *n* connections between each of the four pairs of adjacent switches. This gives a maximum of *4n* connections.
b) We can *n* connections passing through the switch in the upper-right-hand corner and another *n* connections passing through the switch in the lower-left-hand corner, giving a total of *2n* connections.

Problem 4.

Tollbooths are 100 km apart, and the cars propagate at 100km/hr. A tollbooth services a car at a rate of one car every 12 seconds.
a) There are ten cars. It takes 120 seconds, or two minutes, for the first tollbooth to service the 10 cars. Each of these cars has a propagation delay of 60 minutes before arriving at the second tollbooth. Thus, all the cars are lined up before the second tollbooth after 62 minutes. The whole process repeats itself for traveling between the second and third tollbooths. Thus the total delay is 124 minutes.
b) Delay between tollbooths is 7*12 seconds plus 60 minutes, i.e., 61 minutes and 24 seconds. The total delay is twice this amount, i.e., 122 minutes and 48 seconds.

Problem 5

a) $d_{prop} = m/s$ seconds.
b) $d_{trans} = L/R$ seconds.
c) $d_{end-to-end} = (m/s + L/R)$ seconds.
d) The bit is just leaving Host A.
e) The first bit is in the link and has not reached Host B.
f) The first bit has reached Host B.
g) Want

$$m = \frac{L}{R} S = \frac{100}{28 \times 10^3} \left(2.5 \times 10^8\right) = 893 \text{ km.}$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 6

Consider the first bit in a packet. Before this bit can be transmitted, all of the bits in the packet must be generated. This requires

$$\frac{48 \cdot 8}{64 \times 10^3} \text{ sec=6msec.}$$

The time required to transmit the packet is

$$\frac{48 \cdot 8}{1 \times 10^6} \text{ sec=} 384\mu \text{ sec.}$$

Propagation delay = 2 msec.
The delay until decoding is

$$6\text{msec} + 384\mu \text{ sec} + 2\text{msec} = 8.384\text{msec}$$

A similar analysis shows that all bits experience a delay of 8.384 msec.

Problem 7

a) 10 users can be supported because each user requires one tenth of the bandwidth.
b) $p = 0.1$.
c) $\binom{40}{n} p^n (1-p)^{40-n}$.
d) $1 - \sum_{n=0}^{9} \binom{40}{n} p^n (1-p)^{40-n}$.
We use the central limit theorem to approximate this probability. Let X_j be independent random variables such that $P(X_j = 1) = p$.

$$P(\text{ “11 or more users” }) = 1 - P\left(\sum_{j=1}^{40} X_j \le 10\right)$$

$$P\left(\sum_{j=1}^{40} X_j \le 10\right) = P\left(\frac{\sum_{j=1}^{40} X_j - 4}{\sqrt{40 \cdot 0.1 \cdot 0.9}} \le \frac{6}{\sqrt{40 \cdot 0.1 \cdot 0.9}}\right)$$
$$\approx P\left(Z \le \frac{6}{\sqrt{3.6}}\right) = P(Z \le 3.16)$$
$$= 0.999$$

when Z is a standard normal r.v. Thus $P(\text{ “10 or more users” }) \approx 0.001$.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 8

a) 10,000

b) $\sum_{n=N+1}^{M} \binom{M}{n} p^n (1-p)^{M-n}$

Problem 9

The first end system requires L/R_1 to transmit the packet onto the first link; the packet propagates over the first link in d_1/s_1; the packet switch adds a processing delay of d_{proc}; after receiving the entire packet, the packet switch requires L/R_2 to transmit the packet onto the second link; the packet propagates over the second link in d_2/s_2. Adding these five delays gives

$$d_{end\text{-}end} = L/R_1 + L/R_2 + d_1/s_1 + d_2/s_2 + d_{proc}$$

To answer the second question, we simply plug the values into the equation to get 8 + 8 + 16 + 4 + 1 = 37 msec.

Problem 10

Because bits are immediately transmitted, the packet switch does not introduce any delay; in particular, it does not introduce a transmission delay. Thus,

$$d_{end\text{-}end} = L/R + d_1/s_1 + d_2/s_2$$

For the values in Problem 9, we get 8 + 16 + 4 = 28 msec.

Problem 11

The arriving packet must first wait for the link to transmit 3,500 bytes or 28,000 bits. Since these bits are transmitted at 1 Mbps, the queuing delay is 28 msec. Generally, the queuing delay is $[nL + (L - x)]/R$.

Problem 12

The queuing delay is 0 for the first transmitted packet, L/R for the second transmitted packet, and generally, $(n-1)L/R$ for the nth transmitted packet. Thus, the average delay for the N packets is

$$(L/R + 2L/R + \ldots\ldots + (N-1)L/R)/N = L/RN(1 + 2 + \ldots\ldots + (N-1)) = LN(N-1)/(2RN)$$
$$= (N-1)L/(2R)$$

Note that here we used the well-known fact that

$$1 + 2 + \ldots\ldots + N = N(N+1)/2$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 13

It takes LN/R seconds to transmit the N packets. Thus, the buffer is empty when a batch of N packets arrive.

The first of the N packets has no queuing delay. The 2nd packet has a queuing delay of L/R seconds. The nth packet has a delay of $(n-1)L/R$ seconds.

The average delay is

$$\frac{1}{N}\sum_{n=1}^{N}(n-1)L/R = \frac{L}{R}\frac{1}{N}\sum_{n=0}^{N-1}n = \frac{L}{R}\frac{1}{N}\frac{(N-1)N}{2} = \frac{L}{R}\frac{(N-1)}{2}.$$

Problem 14

a) The transmission delay is L/R. The total delay is

$$\frac{IL}{R(1-I)} + \frac{L}{R} = \frac{L/R}{1-I}$$

b) Let $x = L/R$.

$$\text{Total delay} = \frac{x}{1-ax}$$

Problem 15

a) There are Q nodes (the source host and the $N-1$ routers). Let d^q_{proc} denote the processing delay at the q th node. Let R^q be the transmission rate of the q th link and let $d^q_{trans} = L/R^q$. Let d^q_{prop} be the propagation delay across the q th link. Then

$$d_{end-to-end} = \sum_{q=1}^{Q}\left[d^q_{proc} + d^q_{trans} + d^q_{prop}\right].$$

b) Let d^q_{queue} denote the average queueing delay at node q. Then

$$d_{end-to-end} = \sum_{q=1}^{Q}\left[d^q_{proc} + d^q_{trans} + d^q_{prop} + d^q_{queue}\right].$$

Problem 16

The command:

```
traceroute -q 20 www.eurecom.fr
```

will get 20 delay measurements from the issuing host to the host, www.eurecom.fr. The average and standard deviation of these 20 measurements can then be collected. Do you see any differences in your answers as a function of time of day?

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 17

Throughput = $min\{R_s, R_c, R/M\}$

Problem 18

a) 40,000 bits
b) 40,000 bits
c) The bandwidth-delay product of a link is the maximum number of bits that can be in the link
d) 1 bit is 250 meters long, which is longer than a football field
e) s/R

Problem 19

25 bps

Problem 20

a) 40,000,000 bits
b) 400,000 bits
c) .25 meters

Problem 21

a) $t_{trans} + t_{prop}$ = 400 msec + 40 msec = 440 msec
b) 10 * ($t_{trans} + 2\ t_{prop}$) = 10*(40 msec + 80 msec) = 1.2 sec

Problem 22

a) 150 msec
b) 1,500,000 bits
c) 600,000,000 bits

Problem 23

Let's suppose the passenger and his/her bags correspond to the data unit arriving to the top of the protocol stack. When the passenger checks in, his/her bags are checked, and a tag is attached to the bags and ticket. This is additional information added in the Baggage layer if Figure 1.20 that allows the Baggage layer to implement the service or separating the passengers and baggage on the sending side, and then reuniting them (hopefully!) on the destination side. When a passenger then passes through security, and additional stamp is often added to his/her ticket, indicating that the passenger has passed

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

through a security check. This information is used to ensure (e.g., by later checks for the security information) secure transfer of people.

Problem 24

a) Time to send message from source host to first packet switch = $\frac{7.5\times10^6}{1.5\times10^6}\sec = 5\sec$. With store-and-forward switching, the total time to move message from source host to destination host = $5\sec\times 3\ hops = 15\sec$
b) Time to send 1st packet from source host to first packet switch = . $\frac{1.5\times10^3}{1.5\times10^6}\sec = 1\ m\sec$. Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = $2\times1\ m\sec = 2\ m\sec$
c) Time at which 1st packet is received at the destination host = . $1\ m\sec\times 3\ hops = 3\ m\sec$. After this, every 1msec one packet will be received; thus time at which last (5000th) packet is received = $3\ m\sec + 4999 * 1m\sec = 5.002\sec$. It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).
d) Drawbacks:
    i. Packets have to be put in sequence at the destination.
    ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

Problem 25

Java Applet

Problem 26

Time at which the 1st packet is received at the destination = $\frac{S+40}{R}\times 2$ sec. After this, one packet is received at destination every $\frac{S+40}{R}$ sec. Thus delay in sending the whole file =

$$delay = \frac{S+40}{R}\times 2 + (\frac{F}{S}-1)\times(\frac{S+40}{R}) = \frac{S+40}{R}\times(\frac{F}{S}+1)$$

To calculate the value of S which leads to the minimum delay,

$$\frac{d}{dS}delay = 0 \Rightarrow \frac{F}{R}(\frac{1}{S}-\frac{40+S}{S^2})+\frac{1}{R} = 0 \Rightarrow S = \sqrt{40F}$$

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 2 Review Questions

1. The Web: HTTP; file transfer: FTP; remote login: Telnet; Network News: NNTP; e-mail: SMTP.

2. Network architecture refers to the organization of the communication process into layers (e.g., the five-layer Internet architecture). Application architecture, on the other hand, is designed by an application developer and dictates the broad structure of the application (e.g., client-server or P2P)

3. The process which initiates the communication is the client; the process that waits to be contacted is the server.

4. No. As stated in the text, all communication sessions have a client side and a server side. In a P2P file-sharing application, the peer that is receiving a file is typically the client and the peer that is sending the file is typically the server.

5. The IP address of the destination host and the port number of the destination socket.

6. You would use UDP. With UDP, the transaction can be completed in one roundtrip time (RTT) - the client sends the transaction request into a UDP socket, and the server sends the reply back to the client's UDP socket. With TCP, a minimum of two RTTs are needed - one to set-up the TCP connection, and another for the client to send the request, and for the server to send back the reply.

7. There are no good examples of an application that requires no data loss and timing. If you know of one, send an e-mail to the authors.

8. a) Reliable data transfer
   TCP provides a reliable byte-stream between client and server but UDP does not.

   b) A guarantee that a certain value for throughput will be maintained
   Neither

   c) A guarantee that data will be delivered within a specified amount of time
   Neither

   d) Security
   Neither

9. SSL operates at the application layer. The SSL socket takes unencrypted data from the application layer, encrypts it and then passes it to the TCP socket. If the application developer wants TCP to be enhanced with SSL, she has to include the SSL code in the application.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

10. A protocol uses handshaking if the two communicating entities first exchange control packets before sending data to each other. SMTP uses handshaking at the application layer whereas HTTP does not.

11. The applications associated with those protocols require that all application data be received in the correct order and without gaps. TCP provides this service whereas UDP does not.

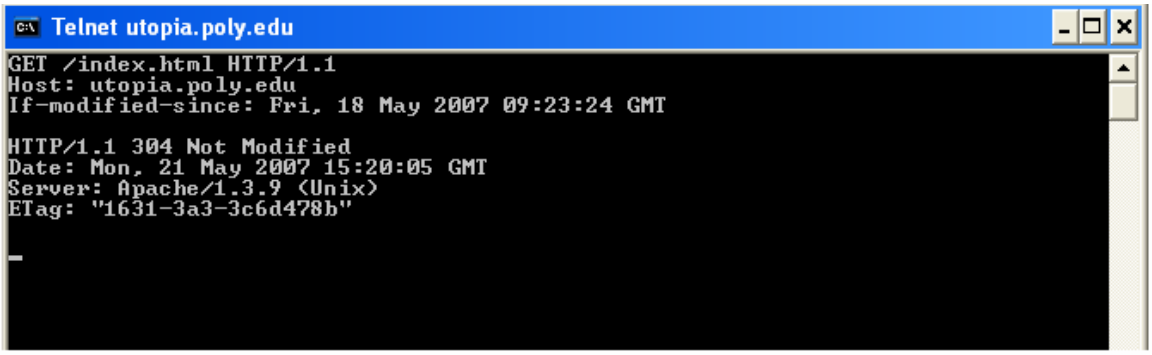
12. When the user first visits the site, the site returns a cookie number. This cookie number is stored on the user's host and is managed by the browser. During each subsequent visit (and purchase), the browser sends the cookie number back to the site. Thus the site knows when this user (more precisely, this browser) is visiting the site.

13. Web caching can bring the desired content "closer" to the user, perhaps to the same LAN to which the user's host is connected. Web caching can reduce the delay for all objects, even objects that are not cached, since caching reduces the traffic on links.

14. Issued the following command (in Windows command prompt) followed by the HTTP GET message to the "utopia.poly.edu" web server:

    > telnet utopia.poly.edu 80

    Since the index.html page in this web server was not modified since Fri, 18 May 2007 09:23:34 GMT, the following output was displayed when the above commands were issued on Sat, 19 May 2007. Note that the first 4 lines are the GET message and header lines input by the user and the next 4 lines (starting from HTTP/1.1 304 Not Modified) is the response from the web server.



15. FTP uses two parallel TCP connections, one connection for sending control information (such as a request to transfer a file) and another connection for actually transferring the file. Because the control information is not sent over the same connection that the file is sent over, FTP sends control information out of band.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

16. Message is sent from Alice's host to her mail server over HTTP. Alice's mail server then sends the message to Bob's mail server over SMTP. Bob then transfers the message from his mail server to his host over POP3.

17.

Received:	from 65.54.246.203 (EHLO bay0-omc3-s3.bay0.hotmail.com) (65.54.246.203) by mta419.mail.mud.yahoo.com with SMTP; Sat, 19 May 2007 16:53:51 -0700
Received:	from hotmail.com ([65.55.135.106]) by bay0-omc3-s3.bay0.hotmail.com with Microsoft SMTPSVC(6.0.3790.2668); Sat, 19 May 2007 16:52:42 -0700
Received:	from mail pickup service by hotmail.com with Microsoft SMTPSVC; Sat, 19 May 2007 16:52:41 -0700
Message-ID:	<BAY130-F26D9E35BF59E0D18A819AFB9310@phx.gbl>
Received:	from 65.55.135.123 by by130fd.bay130.hotmail.msn.com with HTTP; Sat, 19 May 2007 23:52:36 GMT
From:	"prithula dhungel" <prithuladhungel@hotmail.com>
To:	prithula@yahoo.com
Bcc:	
Subject:	Test mail
Date:	Sat, 19 May 2007 23:52:36 +0000
Mime-Version:	1.0
Content-Type:	Text/html; format=flowed
Return-Path:	prithuladhungel@hotmail.com

Figure: A sample mail message header

Received: This header field indicates the sequence in which the SMTP servers send and receive the mail message including the respective timestamps.

In this example there are 4 "Received:" header lines. This means the mail message passed through 5 different SMTP servers before being delivered to the receiver's mail box. The last (forth) "Received:" header indicates the mail message flow from the SMTP server of the sender to the second SMTP server in the chain of servers. The sender's SMTP server is at address 65.55.135.123 and the second SMTP server in the chain is by130fd.bay130.hotmail.msn.com.
The third "Received:" header indicates the mail message flow from the second SMTP server in the chain to the third server, and so on.
Finally, the first "Received:" header indicates the flow of the mail message from the forth SMTP server to the last SMTP server (i.e. the receiver's mail server) in the chain.

Message-id: The message has been given this number BAY130-F26D9E35BF59E0D18A819AFB9310@phx.gbl (by bay0-omc3-

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

s3.bay0.hotmail.com. Message-id is a unique string assigned by the mail system when the message is first created.

From: This indicates the email address of the sender of the mail. In the given example, the sender is "prithuladhungel@hotmail.com"

To: This field indicates the email address of the receiver of the mail. In the example, the receiver is "prithula@yahoo.com"

Subject: This gives the subject of the mail (if any specified by the sender). In the example, the subject specified by the sender is "Test mail"

Date: The date and time when the mail was sent by the sender. In the example, the sender sent the mail on 19th May 2007, at time 23:52:36 GMT.

Mime-version: MIME version used for the mail. In the example, it is 1.0.

Content-type: The type of content in the body of the mail message. In the example, it is "text/html".

Return-Path: This specifies the email address to which the mail will be sent if the receiver of this mail wants to reply to the sender. This is also used by the sender's mail server for bouncing back undeliverable mail messages of mailer-daemon error messages. In the example, the return path is "prithuladhungel@hotmail.com".

18. With download and delete, after a user retrieves its messages from a POP server, the messages are deleted. This poses a problem for the nomadic user, who may want to access the messages from many different machines (office PC, home PC, etc.). In the download and keep configuration, messages are not deleted after the user retrieves the messages. This can also be inconvenient, as each time the user retrieves the stored messages from a new machine, all of non-deleted messages will be transferred to the new machine (including very old messages).

19. Yes an organization's mail server and Web server can have the same alias for a host name. The MX record is used to map the mail server's host name to its IP address.

20. It is not necessary that Bob will also provide chunks to Alice. Alice has to be in the top 4 neighbors of Bob for Bob to send out chunks to her; this might not occur even if Alice is provides chunks to Bob throughout a 30-second interval.

21. Alice will get her first chunk as a result of she being selected by one of her neighbors as a result of an "optimistic unchoke," for sending out chunks to her.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

22. The overlay network in a P2P file sharing system consists of the nodes participating in the file sharing system and the logical links between the nodes. There is a logical link (an "edge" in graph theory terms) from node A to node B if there is a semi-permanent TCP connection between A and B. An overlay network does not include routers. With Gnutella, when a node wants to join the Gnutella network, it first discovers ("out of band") the IP address of one or more nodes already in the network. It then sends join messages to these nodes. When the node receives confirmations, it becomes a member of the of Gnutella network. Nodes maintain their logical links with periodic refresh messages.

23. It is a hybrid of client server and P2P architectures:
    a) There is a centralized component (the index) like in the case of a client server system.
    b) Other functions (except the indexing) do not use any kind of central server. This is similar to what exists in a P2P system.

24. Whenever a user, say Alice wants to locate another user in the instant messaging system, say Bob, her node would send out a query message asking for the (unique) *username* (Bob) into the overlay network, which would be flooded into the network in the same manner a query requesting the location of a resource (eg: a file in Gnutella network) is flooded. If the user with the *username* (Bob) is online, as the query reaches the node where the user is online from, the node sends back the query response to the original query source (Alice). In this way, Alice locates the IP address of the node Bob is currently at in the instant messaging system.
    The advantage of such a design is that there is no centralized component in the system and hence no single point of failure. The disadvantage however, is that the flooding of each such request into the overlay network results in huge traffic in the network. One option is to have limited scope query as in Gnutella. However, with this approach, the user might not be located even if she is online.

25. a) User location
    b) NAT traversal

26. a) File Distribution
    b) Instant Messaging
    c) Video Streaming
    d) Distributed Computing

27. With the UDP server, there is no welcoming socket, and all data from different clients enters the server through this one socket. With the TCP server, there is a welcoming socket, and each time a client initiates a connection to the server, a new socket is created. Thus, to support n simultaneous connections, the server would need $n+1$ sockets.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

28. For the TCP application, as soon as the client is executed, it attempts to initiate a TCP connection with the server. If the TCP server is not running, then the client will fail to make a connection. For the UDP application, the client does not initiate connections (or attempt to communicate with the UDP server) immediately upon execution

Chapter 2 Problems

Problem 1

a) F
b) T
c) F
d) F

Problem 2

Access control commands:
USER, PASS, ACT, CWD, CDUP, SMNT, REIN, QUIT.

Transfer parameter commands:
PORT, PASV, TYPE STRU, MODE.

Service commands:
RETR, STOR, STOU, APPE, ALLO, REST, RNFR, RNTO, ABOR, DELE, RMD, MRD, PWD, LIST, NLST, SITE, SYST, STAT, HELP, NOOP.

Problem 3

Application layer protocols: DNS and HTTP
Transport layer protocols: UDP for DNS; TCP for HTTP

Problem 4

a) The document request was http://gaia.cs.umass.edu/cs453/index.html. The Host : field indicates the server's name and /cs453/index.html indicates the file name.

b) The browser is running HTTP version 1.1,as indicated just before hte first <cr><lf> pair.

c) The browser is requesting a persistent connection, as indicated by the Connection: keep-alive.

d) This is a trick question. This information is not contained in an HTTP message anywhere. So there is no way to tell this from looking at the exchange of HTTP messages alone. One would need information from the IP datagrams (that carried the TCP segment

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

that carried the HTTP GET request) to answer this question.

Problem 5

a) The status code of 200 and the phrase OK indicate that the server was able to locate the document successfully. The reply was provided on Tuesday, 07 Mar 2006 12:39:45 Greenwich Mean Time.

b) The document index.html was last modified on Saturday 10 Dec 2005 18:27:46 GMT.

c) There are 3874 bytes in the document being returned.

d) The first five bytes of the returned document are : <!doc. The server agreed to a persistent connection, as indicated by the Connection: Keep-Alive field

Problem 6

a) Persistent connections are discussed in section 8 of RFC 2616 (the real goal of this question was to get you to retrieve and read an RFC). Sections 8.1.2 and 8.1.2.1 of the RFC indicate that either the client or the server can indicate to the other that it is going to close the persistent connection. It does so by including the including the connection-token "close" in the Connection-header field of the http request/reply.
b) HTTP does not provide any encryption services.

Problem 7

The total amount of time to get the IP address is

$$RTT_1 + RTT_2 + \Lambda + RTT_n .$$

Once the IP address is known, RTT_O elapses to set up the TCP connection and another RTT_O elapses to request and receive the small object. The total response time is

$$2RTT_o + RTT_1 + RTT_2 + \Lambda + RTT_n$$

Problem 8

a)

$$\begin{aligned} & RTT_1 + \Lambda + RTT_n + 2RTT_o + 3 \cdot 2RTT_o \\ = & 8RTT_o + RTT_1 + \Lambda + RTT_n . \end{aligned}$$

b)

$$\begin{aligned} & RTT_1 + \Lambda + RTT_n + 2RTT_o + 2RTT_o \\ = & 4RTT_o + RTT_1 + \Lambda + RTT_n . \end{aligned}$$

c)

$$RTT_1 + \Lambda + RTT_n + 2RTT_o + RTT_o$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

$$= 3RTT_o + RTT_1 + \Lambda + RTT_n .$$

Problem 9

a) The time to transmit an object of size *L* over a link or rate *R* is *L/R*. The average time is the average size of the object divided by *R*:

Δ= (900,000 bits)/(1,500,000 bits/sec) = .6 sec

The traffic intensity on the link is (1.5 requests/sec)(.6 msec/request) = .9. Thus, the average access delay is (.6 sec)/(1 - .9) = 6 seconds. The total average response time is therefore 6 sec + 2 sec = 8 sec.

b) The traffic intensity on the access link is reduced by 40% since the 40% of the requests are satisfied within the institutional network. Thus the average access delay is (.6 sec)/[1 – (.6)(.9)] = 1.2 seconds. The response time is approximately zero if the request is satisfied by the cache (which happens with probability .4); the average response time is 1.2 sec + 2 sec = 3.2 sec for cache misses (which happens 60% of the time). So the average response time is (.4)(0 sec) + (.6)(3.2 sec) = 1.92 seconds. Thus the average response time is reduced from 8 sec to 1.92 sec.

Problem 10

Parallel download would only share the 100K bandwidth among the 10 connections (each getting just 10K bits/sec) thus, there is no significant advantage here. With persistent HTTP we avoid the SYN and SYNACK exchange but that only requires 2 seconds (1 second to send the 100 bit SYN message over the 100 bps link, and 1 second to receive the ACK). Given that each object takes 101 seconds to send and receive the ACK, the use of pipelining gives only a 2 percent gain.

Problem 11

TCPServer.java

```
import java.io.*;
import java.net.*;

class TCPServer {
	public static void main(String argv[]) throws Exception
		{
			String clientSentence;
			ServerSocket welcomeSocket = new ServerSocket(6789);
			while(true) {
				Socket connectionSocket = welcomeSocket.accept();
```

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

```
                    BufferedReader inFromClient = new BufferedReader(new
                    InputStreamReader(connectionSocket.getInputStream( ) ) );

                    clientSentence = inFromClient.readLine();

                    System.out.println("RECEIVED FROM CLIENT : " +
                    clientSentence  + "\n");
                }
            }
}
```

Problem 12

The MAIL FROM: in SMTP is a message from the SMTP client that identifies the sender of the mail message to the SMTP server. The From: on the mail message itself is NOT an SMTP message, but rather is just a line in the body of the mail message.

Problem 13

UIDL abbreviates "unique-ID listing". When a POP3 client issues the UIDL command, the server responds with the unique message ID for all of the messages present in the users mailbox. This command is useful for "download and keep". By keeping a file that lists the messages retrieved in earlier sessions, the client can use the UIDL command to determine which messages on the server have already been seen.

Problem 14

a) **C:** dele 1
C: retr 2
S: (blah blah …
S: ……..blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

b) **C:** retr 2
S: blah blah …
S: ……..blah
S: .
C: quit
S: +OK POP3 server signing off

c) **C:** list
S: 1 498
S: 2 912

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

S: .
C: retr 1
S: blah …..
S: ….blah
S: .
C: retr 2
S: blah blah …
S: ……..blah
S: .
C: quit
S: +OK POP3 server signing off

Problem 15

a) For a given input of domain name (such as ccn.com), IP address or network administrator name, *whois* database can be used to locate the corresponding registrar, whois server, DNS server, and so on.

b) NS4.YAHOO.COM from www.register.com; NS1.MSFT.NET from ww.register.com

c) *Local Domain: www.mindspring.com*
Web servers : www.mindspring.com
207.69.189.21, 207.69.189.22,
207.69.189.23, 207.69.189.24,
207.69.189.25, 207.69.189.26, 207.69.189.27,
207.69.189.28
Mail Servers : mx1.mindspring.com (207.69.189.217)
mx2.mindspring.com (207.69.189.218)
mx3.mindspring.com (207.69.189.219)
mx4.mindspring.com (207.69.189.220)
Name Servers: itchy.earthlink.net (207.69.188.196)
scratchy.earthlink.net (207.69.188.197)

www.yahoo.com
Web Servers: www.yahoo.com (216.109.112.135, 66.94.234.13)
Mail Servers: a.mx.mail.yahoo.com (209.191.118.103)
b.mx.mail.yahoo.com (66.196.97.250)
c.mx.mail.yahoo.com (68.142.237.182, 216.39.53.3)
d.mx.mail.yahoo.com (216.39.53.2)
e.mx.mail.yahoo.com (216.39.53.1)
f.mx.mail.yahoo.com (209.191.88.247, 68.142.202.247)
g.mx.mail.yahoo.com (209.191.88.239, 206.190.53.191)
Name Servers: ns1.yahoo.com (66.218.71.63)
ns2.yahoo.com (68.142.255.16)
ns3.yahoo.com (217.12.4.104)
ns4.yahoo.com (68.142.196.63)

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

ns5.yahoo.com (216.109.116.17)
ns8.yahoo.com (202.165.104.22)
ns9.yahoo.com (202.160.176.146)

www.hotmail.com
Web Servers: www.hotmail.com (64.4.33.7, 64.4.32.7)

Mail Servers: mx1.hotmail.com (65.54.245.8, 65.54.244.8, 65.54.244.136)
mx2.hotmail.com (65.54.244.40, 65.54.244.168, 65.54.245.40)
mx3.hotmail.com (65.54.244.72, 65.54.244.200, 65.54.245.72)
mx4.hotmail.com (65.54.244.232, 65.54.245.104, 65.54.244.104)

Name Servers: ns1.msft.net (207.68.160.190)
ns2.msft.net (65.54.240.126)
ns3.msft.net (213.199.161.77)
ns4.msft.net (207.46.66.126)
ns5.msft.net (65.55.238.126)

d) The yahoo web server has multiple IP addresses
www.yahoo.com (216.109.112.135, 66.94.234.13)

e) The address range for Polytechnic University: 128.238.0.0 – 128.238.255.255

f) An attacker can use the *whois* database and nslookup tool to determine the IP address ranges, DNS server addresses, etc., for the target institution.

g) By analyzing the source address of attack packets, the victim can use whois to obtain information about domain from which the attack is coming and possibly inform the administrators of the origin domain.

Problem 16

For calculating the minimum distribution time for client-server distribution, we use the following formula:

$$D_{cs} = max\ \{NF/u_s,\ F/d_{min}\}$$

Similarly, for calculating the minimum distribution time for P2P distribution, we use the following formula:

$$D_{P2P} = max\{F/u_s,\ F/d_{min},\ NF/(u_s + \sum_{i=1}^{N} u_i)\}$$

Where, F = 10 Gbits = 10 * 1024 Mbits
u_s = 20 Mbps
$d_{min} = d_i$ = 1 Mbps

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Client Server

		N		
		10	**100**	**1000**
	200 Kbps	10240	51200	512000
u	**600 Kbps**	10240	51200	512000
	1 Mbps	10240	51200	512000

Peer to Peer

		N		
		10	**100**	**1000**
	200 Kbps	10240	25904.3	47559.33
u	**600 Kbps**	10240	13029.6	16899.64
	1 Mbps	10240	10240	10240

Problem 17

a) Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of a rate of u_s/N. Note that this rate is less than each of the client's download rate, since by assumption $u_s/N \leq d_{\min}$. Thus each client can also receive at rate u_s/N. Since each client receives at rate u_s/N, the time for each client to receive the entire file is F/(u_s/N) = NF/ u_s. Since all the clients receive the file in NF/ u_s, the overall distribution time is also NF/ u_s.

b) Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of $d_{\min}$. Note that the aggregate rate, $N\, d_{\min}$, is less than the server's link rate u_s, since by assumption $u_s/N \geq d_{\min}$. Since each client receives at rate $d_{\min}$, the time for each client to receive the entire file is F/ $d_{\min}$. Since all the clients receive the file in this time, the overall distribution time is also F/ $d_{\min}$.

c) From Section 2.6 we know that

$D_{CS} \geq \max\ \{NF/u_s,\ F/d_{\min}\}$ (Equation 1)

Suppose that $u_s/N \leq d_{\min}$. Then from Equation 1 we have $D_{CS} \geq NF/u_s$. But from (a) we have $D_{CS} \leq NF/u_s$. Combining these two gives:

$D_{CS} = NF/u_s$ when $u_s/N \leq d_{\min}$. (Equation 2)

We can similarly show that:

$D_{CS} = F/d_{\min}$ when $u_s/N \geq d_{\min}$ (Equation 3).

Combining Equation 2 and Equation 3 gives the desired result.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 18

a) Define $u = u_1 + u_2 + \ldots + u_N$. By assumption

$u_s <= (u_s + u)/N$ Equation 1

Divide the file into N parts, with the ith part having size $(u_i/u)F$. The server transmits the ith part to peer i at rate $r_i = (u_i/u)u_s$. Note that $r_1 + r_2 + \ldots + r_N = u_s$, so that the aggregate server rate does not exceed the link rate of the server. Also have each peer i forward the bits it receives to each of the N-1 peers at rate r_i. The aggregate forwarding rate by peer i is $(N-1)r_i$. We have

$(N-1)r_i = (N-1)(u_s u_i)/u <= u_i$,

where the last inequality follows from Equation 1. Thus the aggregate forwarding rate of peer i is less than its link rate u_i.

In this distribution scheme, peer i receives bits at an aggregate rate of

$$r_i + \sum_{j<>i} r_j = u_s$$

Thus each peer receives the file in F/u_s.

b) Again define $u = u_1 + u_2 + \ldots + u_N$. By assumption

$u_s >= (u_s + u)/N$ Equation 2

Let $r_i = u_i/(N-1)$ and
$r_{N+1} = (u_s - u/(N-1))/N$

In this distribution scheme, the file is broken into N+1 parts. The server sends bits from the ith part to the ith peer (i = 1, …., N) at rate r_i. Each peer i forwards the bits arriving at rate r_i to each of the other N-1 peers. Additionally, the server sends bits from the $(N+1)^{st}$ part at rate r_{N+1} to each of the N peers. The peers do not forward the bits from the $(N+1)^{st}$ part.

The aggregate send rate of the server is

$r_1 + \ldots + r_N + N\, r_{N+1} = u/(N-1) + u_s - u/(N-1) = u_s$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Thus, the server's send rate does not exceed its link rate. The aggregate send rate of peer i is

$(N-1)r_i = u_i$

Thus, each peer's send rate does not exceed its link rate.
In this distribution scheme, peer i receives bits at an aggregate rate of

$$r_i + r_{N+1} + \sum_{j<>i} rj = u/(N-1) + (u_s - u/(N-1))/N = (u_s + u)/N$$

Thus each peer receives the file in $NF/(u_s+u)$.
(For simplicity, we neglected to specify the size of the file part for i = 1, …., N+1. We now provide that here. Let $\Delta = (u_s+u)/N$ be the distribution time. For i = 1, …, N, the ith file part is $F_i = r_i \Delta$ bits. The $(N+1)^{st}$ file part is $F_{N+1} = r_{N+1} \Delta$ bits. It is straightforward to show that $F_1 + \ldots + F_{N+1} = F$.)

c) The solution to this part is similar to that of 17 (c). We know from section 2.6 that

$$D_{P2P} >= max\{F/u_s, NF/(u_s + u)\}$$

Combining this with (a) and (b) gives the desired result.

Problem 19

There are *N* nodes in the overlay network. There are *N(N-1)/2* edges.

Problem 20

a) In this case, each of the five Gnutella clients immediately learns that it has one less neighbor. Consider one of these five clients, called, Bob. Suppose Bob has only three neighbors after X drops out. Then Bob needs to establish a TCP connection with another peer. Bob should have a fresh list of active peers; he sequentially contacts peers on this list until one accepts his TCP connection attempt.

b) In this case, Bob does not immediately know that X has departed. Bob will only learn about X's departure when it attempts to send a message (query or ping) to X. When Bob attempts to send a message, Bob's TCP will make several unsuccessful attempts to send the message to B. Bob's TCP will then inform the Gnutella client that X is down. Bob will then try to establish a TCP connection with a new peer (see part (a)) to rebuild a fifth connection.

Problem 21

a) The advantage of sending the QueryHit message directly over a TCP connection from Bob to Alice is that the QueryHit message is routed by the underlying Internet without

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

passing through intermediate peers; thus, the delay in sending the message from Bob to Alice should be substantially less. The disadvantage is that each peer that has a match would ask Alice to open a TCP connection; Alice may therefore have to opens tens or hundreds of TCP connections for a given query. Furthermore, there will be additional complications if Alice is behind a NAT (see Chapter 4).

b) When a QueryHit message enters a peer, the peer records in a table the MessageID along with an identifier of the TCP socket from which the message arrived. When the same peer receives a QueryHit message with the same MessageID, it indexes the table and determines the socket to which it should forward the message.

c) When the Query message reaches Bob, it contains an ordered list of all the IP addresses of the peers the message passed through between Alice and Bob. When Bob sends back a QueryHit message, it would copy the ordered list into the message. When a peer receives the QueryHit message, it can use the list to determine the next peer in the reverse path.

Problem 22

a) Each super-duper peer is responsible for roughly $200^2 = 40{,}000$ nodes. Therefore, we would need about 100 super-duper peers to support 4 million nodes.
b) Each super peer might store the meta-data for all of the files its children are sharing. A super-duper peer might store all of the meta-data that its super-peer children store. An ordinary node would first send a query to its super peer. The super peer would respond with matches and then possibly forward the message to its super-duper peer. The super-duper peer would respond (through the overlay network) with its matches. The super-duper peer may further forward the query to other super-duper peers.

Problem 23

Alice sends her query to at most N neighbors. Each of these neighbors forwards the query to at most $M = N-1$ neighbors. Each of those neighbors forwards the query to at most M neighbors. Thus the maximum number of query messages is

$$N + NM + NM^2 + \ldots + NM^{(K-1)}$$
$$= N(1 + M + M^2 + \ldots + M^{(K-1)})$$
$$= N(1-M^K)/(1-M)$$
$$= N[(N-1)^K - 1]/(N-2)$$

Problem 24

a) If you run TCPClient first, then the client will attempt to make a TCP connection with a non-existent server process. A TCP connection will not be made.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

b) UDPClient doesn't establish a TCP connection with the server. Thus, everything should work fine if you first run UDPClient, then run UDPServer, and then type some input into the keyboard.

c) If you use different port numbers, then the client will attempt to establish a TCP connection with the wrong process or a non-existent process. Errors will occur.

Problem 25

With the original line, UDPClient does not specify a port number when it creates the socket. In this case, the code lets the underlying operating system choose a port number. With the replacement line, when UDPClient is executed, a UDP socket is created with port number 5432 .

UDPServer needs to know the client port number so that it can send packets back to the correct client socket. Glancing at UDPServer, we see that the client port number is not "hard-wired" into the server code; instead, UDPServer determines the client port number by unraveling the datagram it receives from the client (using the .getPort() method). Thus UDP server will work with any client port number, including 5432. UDPServer therefore does not need to be modified.

Before:

Client socket = x (chosen by OS)
Server socket = 9876

After:

Client socket = 5432

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 3 Review Questions

1. a) Call this protocol Simple Transport Protocol (STP). At the sender side, STP accepts from the sending process a chunk of data not exceeding 1196 bytes, a destination host address, and a destination port number. STP adds a four-byte header to each chunk and puts the port number of the destination process in this header. STP then gives the destination host address and the resulting segment to the network layer. The network layer delivers the segment to STP at the destination host. STP then examines the port number in the segment, extracts the data from the segment, and passes the data to the process identified by the port number.

   b) The segment now has two header fields: a source port field and destination port field. At the sender side, STP accepts a chunk of data not exceeding 1192 bytes, a destination host address, a source port number, and a destination port number. STP creates a segment which contains the application data, source port number, and destination port number. It then gives the segment and the destination host address to the network layer. After receiving the segment, STP at the receiving host gives the application process the application data and the source port number.

   c) No, the transport layer does not have to do anything in the core; the transport layer "lives" in the end systems.

2. a) For sending a letter, the family member is required to give the delegate the letter itself, the address of the destination house, and the name of the recipient. The delegate clearly writes the recipient's name on the top of the letter. The delegate then puts the letter in an envelope and writes the address of the destination house on the envelope. The delegate then gives the letter to the planet's mail service. At the receiving side, the delegate receives the letter from the mail service, takes the letter out of the envelope, and takes note of the recipient name written at the top of the letter. The delegate than gives the letter to the family member with this name.

   b) No, the mail service does not have to open the envelope; it only examines the address on the envelope.

3. Source port number y and destination port number x.

4. An application developer may not want its application to use TCP's congestion control, which can throttle the application's sending rate at times of congestion. Often, designers of IP telephony and IP videoconference applications choose to run their applications over UDP because they want to avoid TCP's congestion control. Also, some applications do not need the reliable data transfer provided by TCP.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

5. Since most firewalls are configured to block UDP traffic, using TCP for video and voice traffic lets the traffic though the firewalls.

6. Yes. The application developer can put reliable data transfer into the application layer protocol. This would require a significant amount of work and debugging, however.

7. Yes, both segments will be directed to the same socket. For each received segment, at the socket interface, the operating system will provide the process with the IP addresses to determine the origins of the individual segments.

8. For each persistent connection, the Web server creates a separate "connection socket". Each connection socket is identified with a four-tuple: (source IP address, source port number, destination IP address, destination port number). When host C receives and IP datagram, it examines these four fields in the datagram/segment to determine to which socket it should pass the payload of the TCP segment. Thus, the requests from A and B pass through different sockets. The identifier for both of these sockets has 80 for the destination port; however, the identifiers for these sockets have different values for source IP addresses. Unlike UDP, when the transport layer passes a TCP segment's payload to the application process, it does not specify the source IP address, as this implicitly specified by the socket identifier.

9. Sequence numbers are required for a receiver to find out whether an arriving packet contains new data or is a retransmission.

10. To handle losses in the channel. If the ACK for a transmitted packet is not received within the duration of the timer for the packet, the packet (or its ACK or NACK) is assumed to have been lost. Hence, the packet is retransmitted.

11. A timer would still be necessary in the protocol rdt 3.0. If the round trip time is known then the only advantage will be that, the sender knows for sure that either the packet or the ACK (or NACK) for the packet has been lost, as compared to the real scenario, where the ACK (or NACK) might still be on the way to the sender, after the timer expires. However, to detect the loss, for each packet, a timer of constant duration will still be necessary at the sender.

12. Java Applet

13. Java Applet

14. a) false b) false c) true d) false e) true f) false g) false

15. a) 20 bytes b) ack number = 90

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

16. 3 segments. First segment: seq = 43, ack =80; Second segment: seq = 80, ack = 44; Third segment; seq = 44, ack = 81

17. R/2

18. False, it is set to half of the current value of the congestion window.

Chapter 3 Problems

Problem 1

	source port numbers	destination port numbers
a) $A \rightarrow S$	467	23
b) $B \rightarrow S$	513	23
c) $S \rightarrow A$	23	467
d) $S \rightarrow B$	23	513

e) Yes.
f) No.

Problem 2

Suppose the IP addresses of the hosts A, B, and C are a, b, c, respectively. (Note that a,b,c are distinct.)

To host A: Source port =80, source IP address = b, dest port = 26145, dest IP address = a

To host C, left process: Source port =80, source IP address = b, dest port = 7532, dest IP address = c

To host C, right process: Source port =80, source IP address = b, dest port = 26145, dest IP address = c

Problem 3

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{rcccccccc} & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

One's complement = 1 1 1 0 1 1 1 0.

To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error. All one-bit errors will be detected, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

Problem 4

(a) Adding the two bytes gives 10011101. Taking the one's complement gives 01100010 (b) Adding the two bytes gives 00011110; the one's complement gives 11100001. (c) first byte = 00110101 ; second byte = 01101000.

Problem 5

No, the receiver cannot be absolutely certain that no bit errors have occurred. This is because of the manner in which the checksum for the packet is calculated. If the corresponding bits (that would be added together) of two 16-bit words in the packet were 0 and 1 then even if these get flipped to 1 and 0 respectively, the sum still remains the same. Hence, the 1s complement the receiver calculates will also be the same. This means the checksum will verify even if there was transmission error.

Problem 6

Suppose the sender is in state "Wait for call 1 from above" and the receiver (the receiver shown in the homework problem) is in state "Wait for 1 from below." The sender sends a packet with sequence number 1, and transitions to "Wait for ACK or NAK 1," waiting for an ACK or NAK. Suppose now the receiver receives the packet with sequence number 1 correctly, sends an ACK, and transitions to state "Wait for 0 from below," waiting for a data packet with sequence number 0. However, the ACK is corrupted. When the rdt2.1 sender gets the corrupted ACK, it resends the packet with sequence number 1. However, the receiver is waiting for a packet with sequence number 0 and (as shown in the home work problem) always sends a NAK when it doesn't get a packet with sequence number 0. Hence the sender will always be sending a packet with sequence number 1, and the receiver will always be NAKing that packet. Neither will progress forward from that state.

Problem 7

To best answer this question, consider why we needed sequence numbers in the first place. We saw that the sender needs sequence numbers so that the receiver can tell if a data packet is a duplicate of an already received data packet. In the case of ACKs, the sender does not need this info (i.e., a sequence number on an ACK) to tell detect a duplicate ACK. A duplicate ACK is obvious to the rdt3.0 receiver, since when it has

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

received the original ACK it transitioned to the next state. The duplicate ACK is not the ACK that the sender needs and hence is ignored by the rdt3.0 sender.

Problem 8

The sender side of protocol rdt3.0 differs from the sender side of protocol 2.2 in that timeouts have been added. We have seen that the introduction of timeouts adds the possibility of duplicate packets into the sender-to-receiver data stream. However, the receiver in protocol rdt.2.2 can already handle duplicate packets. (Receiver-side duplicates in rdt 2.2 would arise if the receiver sent an ACK that was lost, and the sender then retransmitted the old data). Hence the receiver in protocol rdt2.2 will also work as the receiver in protocol rdt 3.0.

Problem 9

Suppose the protocol has been in operation for some time. The sender is in state "Wait for call from above" (top left hand corner) and the receiver is in state "Wait for 0 from below". The scenarios for corrupted data and corrupted ACK are shown in Figure 1.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

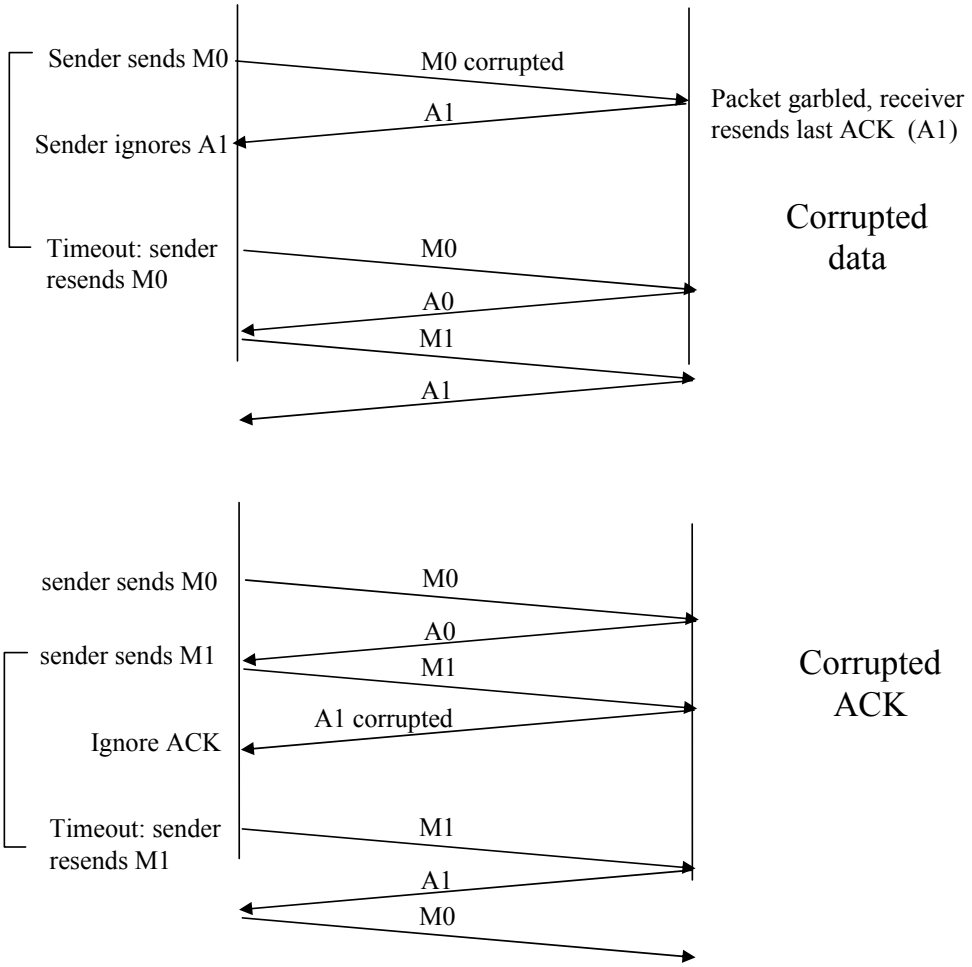


Figure 1: rdt 3.0 scenarios: corrupted data, corrupted ACK

Problem 10

Here, we add a timer, whose value is greater than the known round-trip propagation delay. We add a timeout event to the "Wait for ACK or NAK0" and "Wait for ACK or NAK1" states. If the timeout event occurs, the most recently transmitted packet is retransmitted. Let us see why this protocol will still work with the rdt2.1 receiver.

- Suppose the timeout is caused by a lost data packet, i.e., a packet on the sender-to-receiver channel. In this case, the receiver never received the previous transmission and, from the receiver's viewpoint, if the timeout retransmission is received, it look *exactly* the same as if the original transmission is being received.
- Suppose now that an ACK is lost. The receiver will eventually retransmit the packet on a timeout. But a retransmission is exactly the same action that is take if an ACK is garbled. Thus the sender's reaction is the same with a loss, as with a garbled ACK. The rdt 2.1 receiver can already handle the case of a garbled ACK.

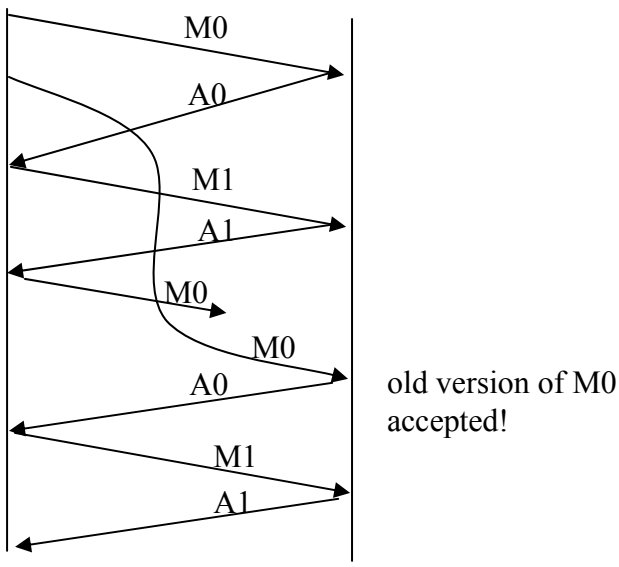
Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 11

The protocol would still work, since a retransmission would be what would happen if the packet received with errors has actually been lost (and from the receiver standpoint, it never knows which of these events, if either, will occur).

To get at the more subtle issue behind this question, one has to allow for premature timeouts to occur. In this case, if each extra copy of the packet is ACKed and each received extra ACK causes another extra copy of the current packet to be sent, the number of times packet n is sent will increase without bound as n approaches infinity.

Problem 12



Problem 13

In a NAK only protocol, the loss of packet x is only detected by the receiver when packet $x+1$ is received. That is, the receivers receives $x-1$ and then $x+1$, only when $x+1$ *is* received does the receiver realize that x was missed. If there is a long delay between the transmission of x and the transmission of $x+1$, then it will be a long time until x can be recovered, under a NAK only protocol.

On the other hand, if data is being sent often, then recovery under a NAK-only scheme could happen quickly. Moreover, if errors are infrequent, then NAKs are only occasionally sent (when needed), and ACK are never sent – a significant reduction in feedback in the NAK-only case over the ACK-only case.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 14

It takes 8 microseconds (or 0.008 milliseconds) to send a packet. in order for the sender to be busy 90 percent of the time, we must have

$$util = 0.9 = (0.008n)/30.016$$

or n approximately 3377 packets.

Problem 15

In our solution, the sender will wait until it receives an ACK for a pair of messages (seqnum and seqnum+1) before moving on to the next pair of messages. Data packets have a data field and carry a two-bit sequence number. That is, the valid sequence numbers are 0, 1, 2, and 3. (Note: you should think about why a 1-bit sequence number space of 0, 1 only would not work in the solution below.) ACK messages carry the sequence number of the data packet they are acknowledging.

The FSM for the sender and receiver are shown in Figure 2. Note that the sender state records whether (i) no ACKs have been received for the current pair, (ii) an ACK for seqnum (only) has been received, or an ACK for seqnum+1 (only) has been received. In this figure, we assume that the seqnum is initially 0, and that the sender has sent the first two data messages (to get things going). A timeline trace for the sender and receiver recovering from a lost packet is shown below:

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

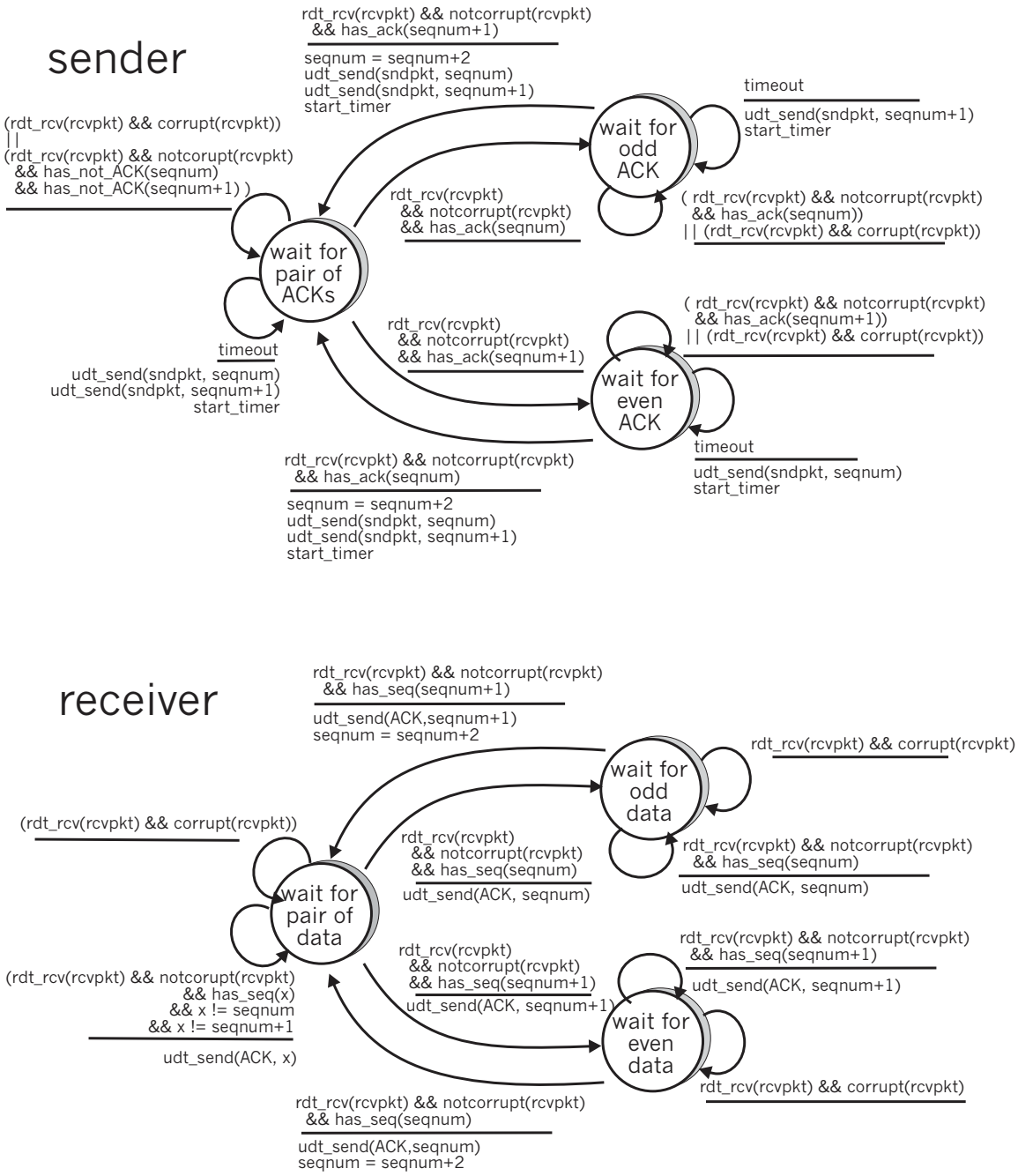


Figure 2: Sender and receiver for Problem 3.15

```
Sender                                  Receiver

make pair (0,1)
send packet 0
                    Packet 0 drops
send packet 1
                                        receive packet 1
                                        buffer packet 1
```

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

```
                                        send ACK 1
receive ACK 1
(timeout)
resend packet 0
                                        receive packet 0
                                        deliver pair (0,1)
                                        send ACK 0
receive ACK 0
```

Problem 16

This problem is a variation on the simple stop and wait protocol (rdt3.0). Because the channel may lose messages and because the sender may resend a message that one of the receivers has already received (either because of a premature timeout or because the other receiver has yet to receive the data correctly), sequence numbers are needed. As in rdt3.0, a 0-bit sequence number will suffice here.

The sender and receiver FSM are shown in Figure 3. In this problem, the sender state indicates whether the sender has received an ACK from B (only), from C (only) or from neither C nor B. The receiver state indicates which sequence number the receiver is waiting for.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

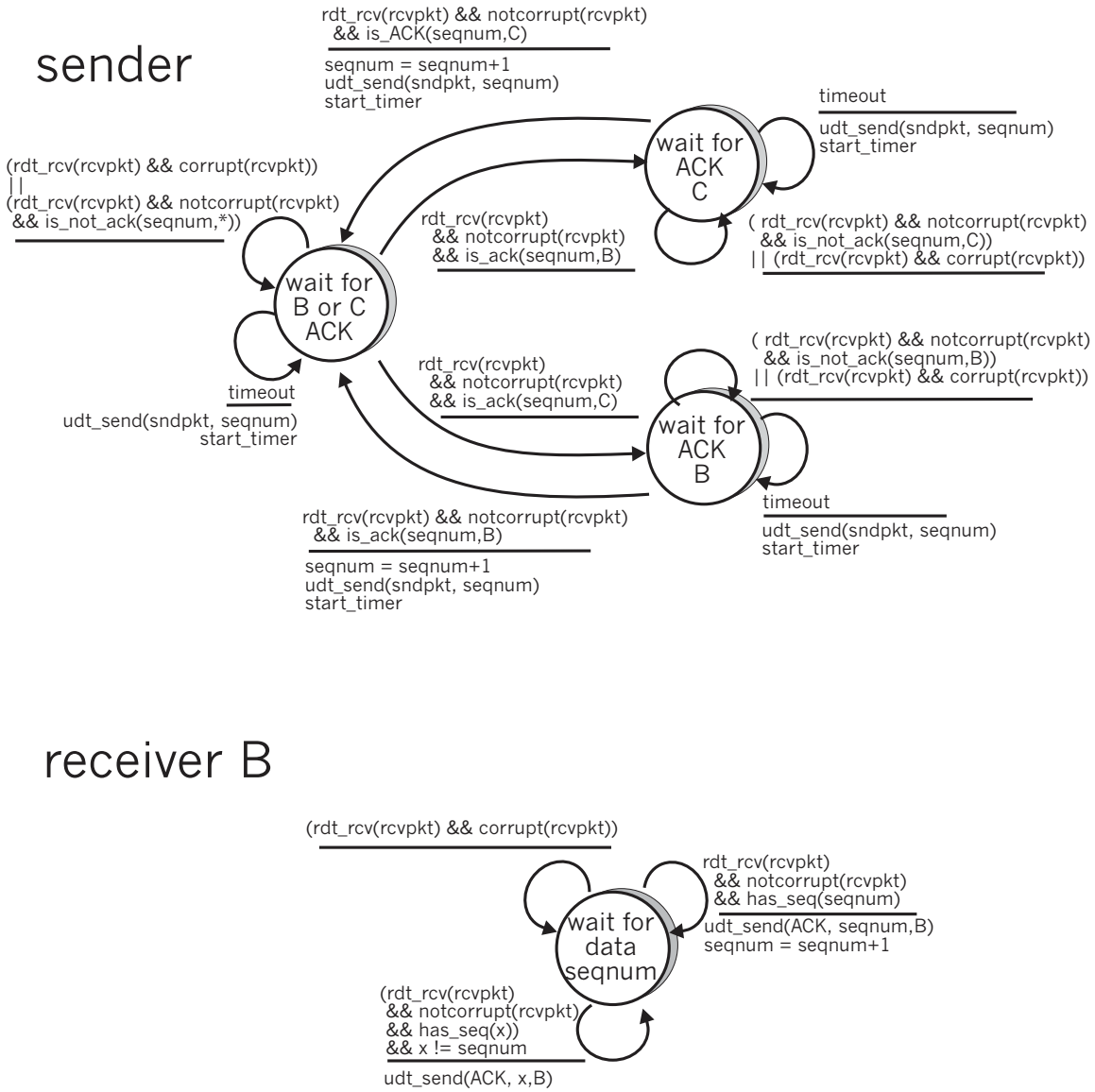


Figure 3. Sender and receiver for Problem 3.16

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 17

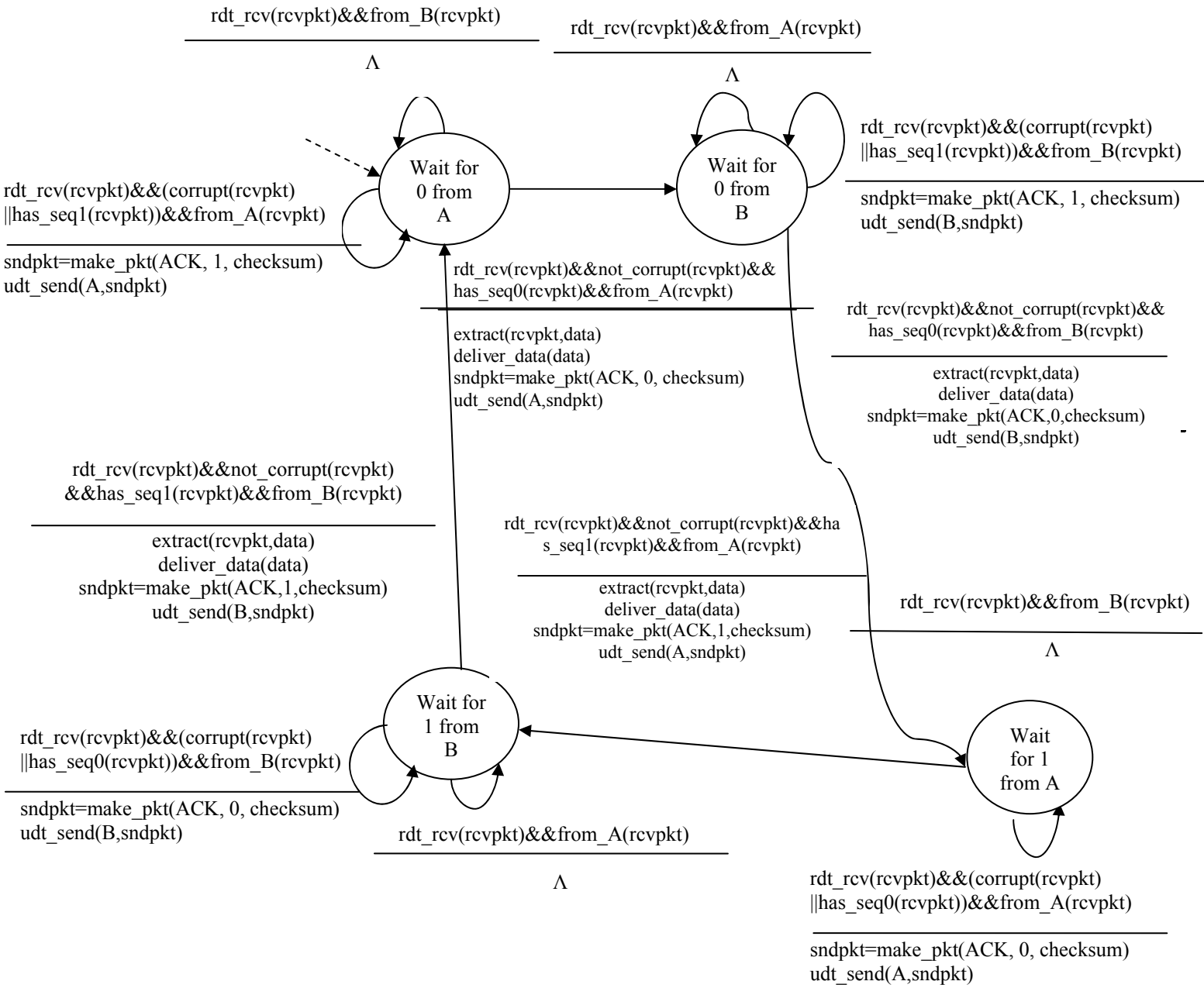


Figure 4: Receiver side FSM for 3.17

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Sender
The sender side FSM is exactly same as given in Figure 3.15 in text

Problem 18

a) Here we have a window size of N=3. Suppose the receiver has received packet k-1, and has ACKed that and all other preceeding packets. If all of these ACK's have been received by sender, then sender's window is [k, k+N-1]. Suppose next that none of the ACKs have been received at the sender. In this second case, the sender's window contains k-1 and the N packets up to and including k-1. The sender's window is thus [k-N,k-1]. By these arguments, the senders window is of size 3 and begins somewhere in the range [k-N,k].

b) If the receiver is waiting for packet k, then it has received (and ACKed) packet k-1 and the N-1 packets before that. If none of those N ACKs have been yet received by the sender, then ACK messages with values of [k-N,k-1] may still be propagating back.
Because the sender has sent packets [k-N, k-1], it must be the case that the sender has already received an ACK for k-N-1. Once the receiver has sent an ACK for k-N-1 it will never send an ACK that is less that k-N-1. Thus the range of in-flight ACK values can range from k-N-1 to k-1.

Problem 19

Because the A-to-B channel can lose request messages, A will need to timeout and retransmit its request messages (to be able to recover from loss). Because the channel delays are variable and unknown, it is possible that A will send duplicate requests (i.e., resend a request message that has already been received by B). To be able to detect duplicate request messages, the protocol will use sequence numbers. A 1-bit sequence number will suffice for a stop-and-wait type of request/response protocol.
A (the requestor) has 4 states:

- **"Wait for Request 0 from above."** Here the requestor is waiting for a call from above to request a unit of data. When it receives a request from above, it sends a request message, R0, to B, starts a timer and makes a transition to the "Wait for D0" state. When in the "Wait for Request 0 from above" state, A ignores anything it receives from B.

- **"Wait for D0"**. Here the requestor is waiting for a D0 data message from B. A timer is always running in this state. If the timer expires, A sends another R0 message, restarts the timer and remains in this state. If a D0 message is received from B, A stops the time and transits to the "Wait for Request 1 from above" state. If A receives a D1 data message while in this state, it is ignored.

- **"Wait for Request 1 from above."** Here the requestor is again waiting for a call from above to request a unit of data. When it receives a request from above, it sends a request message, R1, to B, starts a timer and makes a transition to the

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

"Wait for D1" state. When in the "Wait for Request 1 from above" state, A ignores anything it receives from B.

- **"Wait for D1".** Here the requestor is waiting for a D1 data message from B. A timer is always running in this state. If the timer expires, A sends another R1 message, restarts the timer and remains in this state. If a D1 message is received from B, A stops the timer and transits to the "Wait for Request 0 from above" state. If A receives a D0 data message while in this state, it is ignored.

The data supplier (B) has only two states:

- **"Send D0."** In this state, B continues to respond to received R0 messages by sending D0, and then remaining in this state. If B receives a R1 message, then it knows its D0 message has been received correctly. It thus discards this D0 data (since it has been received at the other side) and then transits to the "Send D1" state, where it will use D1 to send the next requested piece of data.

- **"Send D1."** In this state, B continues to respond to received R1 messages by sending D1, and then remaining in this state. If B receives a R1 message, then it knows its D1 message has been received correctly and thus transits to the "Send D1" state.

Problem 20

In order to avoid the scenario of Figure 3.27, we want to avoid having the leading edge of the receiver's window (i.e., the one with the "highest" sequence number) wrap around in the sequence number space and overlap with the trailing edge (the one with the "lowest" sequence number in the sender's window). That is, the sequence number space must be large enough to fit the entire receiver window and the entire sender window without this overlap condition. So - we need to determine how large a range of sequence numbers can be covered at any given time by the receiver and sender windows.

Suppose that the lowest-sequence number that the receiver is waiting for is packet m. In this case, it's window is [m,m+w-1] and it has received (and ACKed) packet m-1 and the w-1 packets before that, where w is the size of the window. If none of those w ACKs have been yet received by the sender, then ACK messages with values of [m-w,m-1] may still be propagating back. If no ACKs with these ACK numbers have been received by the sender, then the sender's window would be [m-w,m-1].

Thus, the lower edge of the sender's window is m-w, and the leading edge of the receivers window is m+w-1. In order for the leading edge of the receiver's window to not overlap with the trailing edge of the sender's window, the sequence number space must thus be big enough to accommodate 2w sequence numbers. That is, the sequence number space must be at least twice as large as the window size, $k \geq 2w$.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 21

a) True. Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at $t0$. At $t1$ $(t1 > t0)$ the receiver ACKS 1, 2, 3. At $t2$ $(t2 > t1)$ the sender times out and resends 1, 2, 3. At $t3$ the receiver receives the duplicates and re-acknowledges 1, 2, 3. At $t4$ the sender receives the ACKs that the receiver sent at $t1$ and advances its window to 4, 5, 6. At $t5$ the sender receives the ACKs 1, 2, 3 the receiver sent at $t2$. These ACKs are outside its window.

b) True. By essentially the same scenario as in (a).

c) True.

d) True. Note that with a window size of 1, SR, GBN, and the alternating bit protocol are functionally equivalent. The window size of 1 precludes the possibility of out-of-order packets (within the window). A cumulative ACK is just an ordinary ACK in this situation, since it can only refer to the single packet within the window.

Problem 22

a) Consider sending an application message over a transport protocol. With TCP, the application writes data to the connection send buffer and TCP will grab bytes without necessarily putting a single message in the TCP segment; TCP may put more or less than a singe message in a segment. UDP, on the other hand, encapsulates in a segment whatever the application gives it; so that, if the application gives UDP an application message, this message will be the payload of the UDP segment. Thus, with UDP, an application has more control of what data is sent in a segment.

b) With TCP, due to flow control and congestion control, there may be significant delay from the time when an application writes data to its send buffer until when the data is given to the network layer. UDP does not have delays due to flow control and congestion control.

Problem 23

There are $2^{32} = 4{,}294{,}967{,}296$ possible sequence numbers.

a) The sequence number does not increment by one with each segment. Rather, is increments by the number of bytes of data sent. So the size of the MSS is irrelevant -- the maximum size file that can be sent from A to B is simply the number of bytes representable by $2^{32} \approx 4.19\,\text{Gbytes}$.

b) The number of segments is $\left\lceil \frac{2^{32}}{1460} \right\rceil = 2{,}941{,}758$. 66 bytes of header get added to each segment giving a total of 194,156,028 bytes of header. The total number of bytes transmitted is $2^{32} + 194{,}156{,}028 = 3{,}591 \times 10^7$ bits.

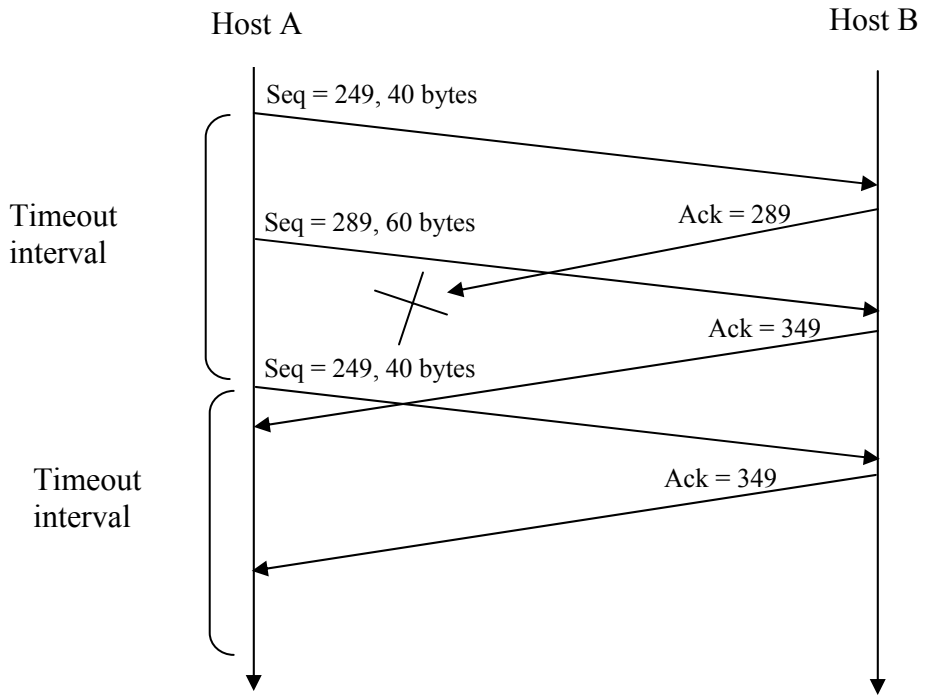
This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Thus it would take 3,591 seconds = 59 minutes to transmit the file over a 10~Mbps link.

Problem 24

a. In the second segment from Host A to B, the sequence number is 289, source port number is 503 and destination port number is 80.
b. If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 289, the source port number is 80 and the destination port number is 503.
c. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 249, indicating that it is still waiting for bytes 249 and onwards.
d.



Problem 25

Host A sends data into the receive buffer faster than Host B can remove data from the buffer. The receive buffer fills up at a rate of roughly 50Mbps. When the buffer is full, Host B signals to Host A to stop sending data by setting RcvWindow = 0. Host A then stops sending until it receives a TCP segment with RcvWindow > 0. Host A will thus repeatedly stop and start sending as a function of the RcvWindow values it receives from Host B. On average, the long-term rate at which Host A sends data to Host B as part of this connection is no more than 50Mbps.

Problem 26

a) The server uses special initial sequence number (that is obtained from the hash of

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

source and destination IPs and ports) in order to defend itself against SYN FLOOD attack.

b) No, the attacker cannot create half-open or fully open connections by simply sending and ACK packet to the target. Half-open connections are not possible since a server using SYN cookies does not maintain connection variables and buffers for any connection before full connections are established. For establishing fully open connections, an attacker should know the special initial sequence number correspoding to the (spoofed) source IP address from the attacker. This sequence number requires the "secret" number that each server uses. Since the attacker does not know this secret number, she cannot guess the initial sequence number.

Problem 27

Denote $EstimatedRTT^{(n)}$ for the estimate after the *n*th sample.

$$EstimatedRTT^{(1)} = SampleRTT_1$$

$$EstimatedRTT^{(2)} = xSampleRTT_1 + (1-x)SampleRTT_2$$

$$\begin{aligned} EstimatedRTT^{(3)} &= xSampleRTT_1 \\ &+ (1-x)[xSampleRTT_2 + (1-x)SampleRTT_3] \\ &= xSampleRTT_1 + (1-x)xSampleRTT_2 \\ &+ (1-x)^2 SampleRTT_3 \end{aligned}$$

$$\begin{aligned} EstimatedRTT^{(4)} &= xSampleRTT_1 + (1-x)EstimatedRTT^{(3)} \\ &= xSampleRTT_1 + (1-x)xSampleRTT_2 \\ &+ (1-x)^2 xSampleRTT_3 + (1-x)^3 SampleRTT_4 \end{aligned}$$

b)

$$\begin{aligned} EstimatedRTT^{(n)} &= x\sum_{j=1}^{n-1}(1-x)^j SampleRTT_j \\ &+ (1-x)^n SampleRTT_n \end{aligned}$$

c)

$$EstimatedRTT^{(\infty)} = \frac{x}{1-x}\sum_{j=1}^{\infty}(1-x)^j SampleRTT_j$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

$$= \frac{1}{9}\sum_{j=1}^{\infty} .9^{j}\, SampleRTT_{j}$$

The weight given to past samples decays exponentially.

Problem 28

Let's look at what could wrong if TCP measures `SampleRTT` for a retransmitted segment. Suppose the source sends packet P1, the timer for P1 expires, and the source then sends P2, a new copy of the same packet. Further suppose the source measures `SampleRTT` for P2 (the retransmitted packet). Finally suppose that shortly after transmitting P2 an acknowledgment for P1 arrives. The source will mistakenly take this acknowledgment as an acknowledgment for P2 and calculate an incorrect value of `SampleRTT`.

Problem 29

At any given time *t*, `SendBase - 1` is the sequence number of the last byte that the sender knows has been received correctly, and in order, at the receiver. The actually last byte received (correctly and in order) at the receiver at time *t* may be greater if there are acknowledgements in the pipe. Thus

`SendBase-1 ≤ LastByteRcvd`

Problem 30

When, at time *t*, the sender receives an acknowledgement with value *y*, the sender knows for sure that the receiver has received everything up through *y*-1. The actual last byte received (correctly and in order) at the receiver at time *t* may be greater if y ≤ `SendBase` or if there are other acknowledgements in the pipe. Thus

`y-1 ≤ LastByteRvcd`

Problem 31

Suppose packets n, n+1, and n+2 are sent, and that packet n is received and ACKed. If packets n+1 and n+2 are reordered along the end-to-end-path (i.e., are received in the order n+2, n+1) then the receipt of packet n+2 will generate a duplicate ack for n and would trigger a retransmission under a policy of waiting only for second duplicate ACK for retransmission. By waiting for a triple duplicate ACK, it must be the case that *two* packets after packet *n* are correctly received, while n+1 was not received. The designers of the triple duplicate ACK scheme probably felt that waiting for two packets (rather than 1) was the right tradeoff between triggering a quick retransmission when needed, but not retransmitting prematurely in the face of packet reordering.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 32

If the arrival rate increases beyond R/2 in Figure 3.46(b), then the total arrival rate to the queue exceeds the queue's capacity, resulting in increasing loss as the arrival rate increases. When the arrival rate equals R/2, 1 out of every three packets that leaves the queue is a retransmission. With increased loss, even a larger fraction of the packets leaving the queue will be retransmissions. Given that the maximum departure rate from the queue for one of the sessions is R/2, and given that a third or more will be transmissions as the arrival rate increases, the throughput of successfully deliver data can not increase beyond λ_{out}. Following similar reasoning, if half of the packets leaving the queue are retransmissions, and the maximum rate of output packets per session is R/2, then the maximum value of λ_{out} is (R/2)/2 or R/4.

Problem 33

a) TCP slowstart is operating in the intervals [1,6] and [23,26]
b) TCP congestion advoidance is operating in the intervals [6,16] and [17,22]
c) After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.
d) After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.
e) The threshold is initially 32, since it is at this window size that slowtart stops and congestion avoidance begins.
f) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.
g) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.
h) During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets15-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.
i) The congestion window and threshold will be set to half the current value of the congestion window (8) when the loss occurred. Thus the new values of the threshold and window will be 4.

Problem 34

Refer to Figure 5. In Figure 5(a), the ratio of the linear decrease on loss between connection 1 and connection 2 is the same - as ratio of the linear increases: unity. In this case, the throughputs never move off of the AB line segment. In Figure 5(a), the ratio of the linear decrease on loss between connection 1 and connection 2 is 2:1. That is,

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

whenever there is a loss, connection 1 decreases its window by twice the amount of connection 2. We see that eventually, after enough losses, and subsequent increases, that connection 1's throughput will go to 0, and the full link bandwidth will be allocated to connection 2.

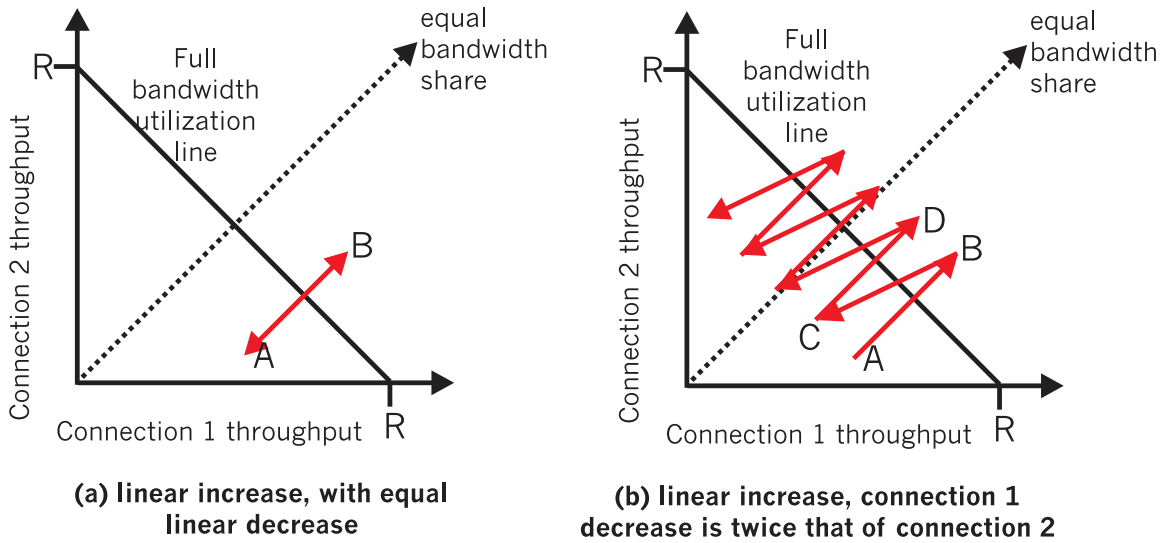


Figure 5: Lack of TCP convergence with linear increase, linear decrease

Problem 35

If TCP were a stop-and-wait protocol, then the doubling of the time out interval would suffice as a congestion control mechanism. However, TCP uses pipelining (and is therefore not a stop-and-wait protocol), which allows the sender to have multiple outstanding unacknowledged segments. The doubling of the timeout interval does not prevent a TCP sender from sending a large number of first-time-transmitted packets into the network, even when the end-to-end path is highly congested. Therefore a congestion-control mechanism is needed to stem the flow of "data received from the application above" when there are signs of network congestion.

Problem 36

In this problem, there is no danger in overflowing the receiver since the receiver's receive buffer can hold the entire file. Also, because there is no loss and acknowledgements are returned before timers expire, TCP congestion control does not throttle the sender. However, the process in host A will not continuously pass data to the socket because the send buffer will quickly fill up. Once the send buffer becomes full, the process will pass data at an average rate or R << S.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 37

a) It takes 1 RTT to increase CongWin to 2 MSS; 2 RTTs to increase to 3 MSS; 3 RTTs to increase to 4 MSS; 4 RTTs to increase to 5 MSS; and 5 RTTs to increase to 6 MSS.

b) In the first RTT 1 MSS was sent; in the second RTT 2 MSS was sent; in the third RTT 3 MSS was sent; in the forth RTT 4 MSS was sent; in the fifth RTT, 5 MSS was sent. Thus, up to time 5 RTT, 1+2+3+4+5 = 15 MSS were sent (and acknowledged). Thus, once can say that the average throughput up to time 5 RTT was (15 MSS)/(5 RTT) = 3 MSS/RTT.

Problem 38

The loss rate, L, is the ratio of the number of packets lost over the number of packets sent. In a cycle, 1 packet is lost. The number of packets sent in a cycle is

$$\frac{W}{2}+\left(\frac{W}{2}+1\right)+\Lambda\ +W=\sum_{n=0}^{W/2}(\frac{W}{2}+n)$$

$$=\left(\frac{W}{2}+1\right)\frac{W}{2}+\sum_{n=0}^{W/2}n$$

$$=\left(\frac{W}{2}+1\right)\frac{W}{2}+\frac{W/2(W/2+1)}{2}$$

$$=\frac{W^2}{4}+\frac{W}{2}+\frac{W^2}{8}+\frac{W}{4}$$

$$=\frac{3}{8}W^2+\frac{3}{4}W$$

Thus the loss rate is

$$L=\frac{1}{\frac{3}{8}W^2+\frac{3}{4}W}$$

b) For W large, $\frac{3}{8}W^2>>\frac{3}{4}W$. Thus $L\approx 8/3W^2$ or $W\approx\sqrt{\frac{8}{3L}}$. From the text, we therefore have

$$\text{average throughput}=\frac{3}{4}\sqrt{\frac{8}{3L}}\cdot\frac{MSS}{RTT}$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

$$= \frac{1.22 \cdot MSS}{RTT \cdot \sqrt{L}}$$

Problem 39

Let's assume 1500-byte packets and a 100 ms round-trip time. From the TCP throughput equation on page 271, we have

10 Gbps = 1.22 * (1500*8 bits) / (.1 sec * srqt(L)), or

sqrt(L) = 14640 bits / (10^^9 bits) = 0.00001464, or

L = 2.14 * 10^^-10

Problem 40

An advantage of using the earlier values of `CongWin` and `Threshold` at t_2 is that TCP would not have to go through slow start and congestion avoidance to ramp up to the throughput value obtained at t_1. A disadvantage of using these values is that they may be no longer accurate. In particular, if the path has become more congested between t^1 and t_2, the sender will send a large window's worth of segments into an already (more) congested path.

Problem 41

a) The server will send its response to Y.

b) No. In this case, if the client spoofs its IP to Y, she cannot complete the three-way handshake with the server and so no further responses will be sent from the server to the spoofed address Y.

c) The server can be certain that the client is indeed at Y. If it were at some other address spoofing Y, the SYNACK would have been sent to the address Y, and the TCP in that host would not send the TCP ACK segment back. Even if the attacker were to send an appropriately timed TCP ACK segment, it would not know the correct server sequence number (since the server uses random initial sequence numbers.)

Problem 42

a) Referring to the figure below, we see that the total delay is

RTT + RTT + S/R + RTT + S/R + RTT + 12S/R = 4RTT + 14 S/R

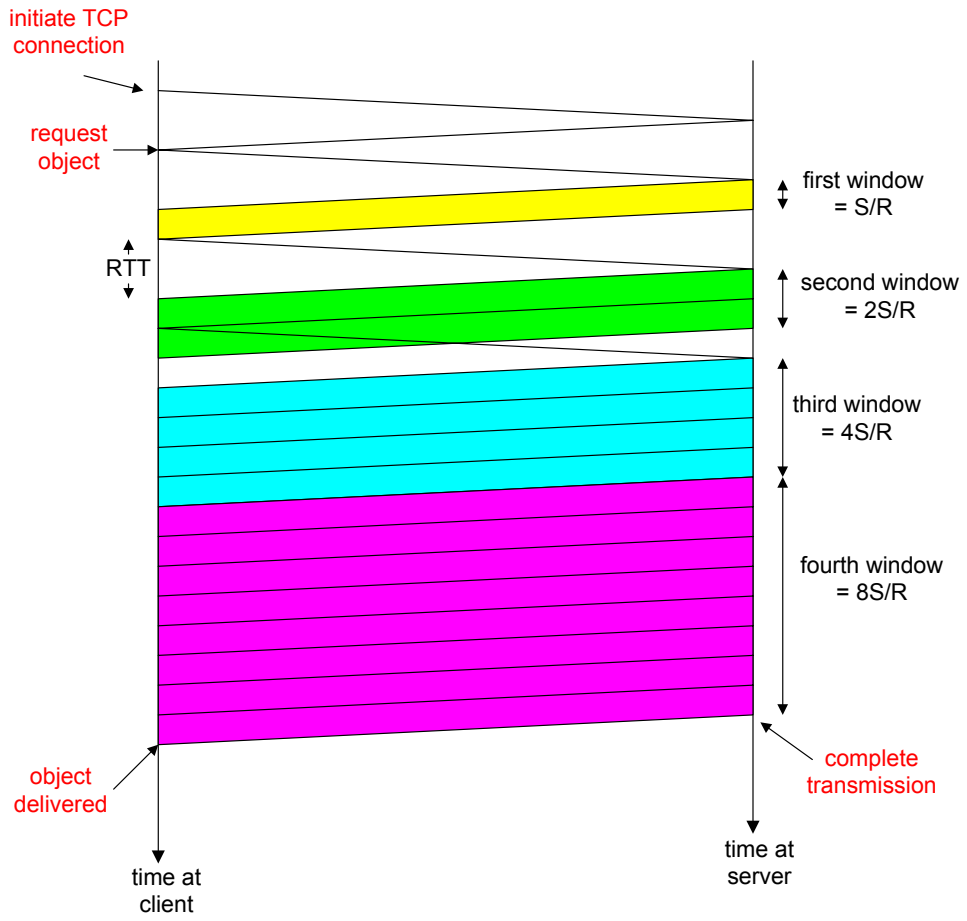
b) Similarly, the delay in this case is:

RTT+RTT + S/R + RTT + S/R + RTT + S/R + RTT + 8S/R = 5RTT +11 S/R

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

c) Similarly, the delay in this case is:

RTT + RTT + S/R + RTT + 14 S/R = 3 RTT + 15 S/R



This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 4 Review Questions

1. A network-layer packet is a datagram. A router forwards a packet based on the packet's IP (layer 3) address. A link-layer switch forwards a packet based on the packet's MAC (layer 2) address.

2. Datagram-based network layer: forwarding; routing. Additional function of VC-based network layer: call setup.

3. Forwarding is about moving a packet from a router's input link to the appropriate output link. Routing is about determining the end-to-routes between sources and destinations.

4. Yes, both use forwarding tables. For descriptions of the tables, see Section 4.2.

5. Single packet: guaranteed delivery; guaranteed delivery with guaranteed maximum delay. Flow of packets: in-order packet delivery; guaranteed minimal bandwidth; guaranteed maximum jitter. None of these services is provided by the Internet's network layer. ATM's CBR service provides both guaranteed delivery and timing. ABR does not provide any of these services.

6. Interactive live multimedia applications, such as IP telephony and video conference, could benefit from ATM CBR's service, which maintains timing.

7. With the shadow copy, the forwarding decision is made locally, at each input port, without invoking the centralized routing processor. Such decentralized forwarding avoids creating a forwarding processing bottleneck at a single point within the router.

8. Switching via memory; switching via a bus; switching via an interconnection network

9. Packet loss occurs if queue size at the input port grows large because of slow switching fabric speed and thus exhausting router's buffer space. It can be eliminated if the switching fabric speed is at least n times as fast as the input line speed, where n is the number of input ports.

10. Packet loss can occur if the queue size at the output port grows large because of slow outgoing line-speed.

11. HOL blocking – a queued packet in an input queue must wait for transfer through the fabric because it is blocked by another packet at the head of the line. It occurs at the input port.

12. Yes. They have one address for each interface.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

13. 11011111 00000001 00000011 00011100

14. Students will get different correct answers for this question.

15. 8 interfaces; 3 forwarding tables

16. 50% overhead

17. The 8-bit protocol field in the IP datagram contains information about which transport layer protocol the destination host should pass the segment to.

18. Typically the wireless router includes a DHCP server. DHCP is used to assign IP addresses to the 5 PCs and to the router interface. Yes, the wireless router also uses NAT as it obtains only one IP address from the ISP.

19. See Section 4.4.4

20. Yes, because the entire IPv6 datagram (including header fields) is encapsulated in an IPv4 datagram

21. Link state algorithms: Computes the least-cost path between source and destination using complete, global knowledge about the network. Distance-vector routing: The calculation of the least-cost path is carried out in an iterative, distributed manner. A node only knows the neighbor to which it should forward a packet in order to reach given destination along the least-cost path, and the cost of that path from itself to the destination.

22. Routers are aggregated into autonomous systems (ASs). Within an AS, all routers run the same intra-AS routing protocol. Special gateway routers in the various ASs run the inter-autonomous system routing protocol that determines the routing paths among the ASs. The problem of scale is solved since an intra-AS router need only know about routers within its AS and the gateway router(s) in its AS.

23. No. Each AS has administrative autonomy for routing within an AS.

24. No. The advertisement tells D that it can get to z in 11 hops by way of A. However, D can already get to z by way of B in 7 hops. Therefore, there is no need to modify the entry for z in the table. If, on the other hand, the advertisement said that A were only 4 hops away from z by way of C, then D would indeed modify its forwarding table.

25. With OSPF, a router periodically broadcasts routing information to all other routers in the AS, not just to its neighboring routers. This routing information sent by a router has one entry for each of the router's neighbors; the entry gives the distance from the router to the neighbor. A RIP advertisement sent by a router

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

contains information about all the networks in the AS, although this information is only sent to its neighboring routers.

26. "sequence of ASs on the routes"

27. See "Principles in Practice" on page 401

28. ISP C can use the BGP Multi-Exit Descriptor to suggest to ISP B that the preferred route to ISP D is through the east coast peering point. For example, the east coast BGP router in ISP C can advertise a route to D with an MED value of 5. The west coast router in ISP C can advertise a route to D with an MED value of 10. Since a lower value is preferred, ISP B knows that ISP C wants to receive traffic on the east coast. In practice, a router can ignore the MED value, and so ISP B can still use hot potato routing to pass traffic to ISP C destined to ISP D via the west coast peering point.

29. A **subnet** is a portion of a larger network; a subnet does not contain a router; its boundaries are defined by the router and host interfaces. A **prefix** is the network portion of a CDIRized address; it is written in the form a.b.c.d/x ; A prefix covers one or more subnets. When a router advertises a prefix across a BGP session, it includes with the prefix a number of BGP attributes. In BGP jargon, a prefix along with its attributes is a **BGP route** (or simply a **route**).

30. Routers use the AS-PATH attribute to detect and prevent looping advertisements; they also use it in choosing among multiple paths to the same prefix. The NEXT-HOP attribute indicates the IP address of the first router along an advertised path (outside of the AS receiving the advertisement) to a given prefix. When configuring its forwarding table, a router uses the NEXT-HOP attribute.

31. A tier-1 ISP B may not to carry transit traffic between two other tier-1 ISPs, say A and C, with which B has peering agreements. To implement this policy, ISP B would not advertise to A routes that pass through C; and would not advertise to C routes that pass through A.

32. N-way unicast has a number of drawbacks, including:

    - Efficiency: multiple copies of the same packet are sent over the same link for potentially many links; source must generate multiple copies of same packet
    - Addressing: the source must discover the address of all the recipients

33. a) uncontrolled flooding: T; controlled flooding: T; spanning-tree: F
    b) uncontrolled flooding: T; controlled flooding: F; spanning-tree: F

34. False

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

35. IGMP is a protocol run only between the host and its first-hop multicast router. IGMP allows a host to specify (to the first-hop multicast router) the multicast group it wants to join. It is then up to the multicast router to work with other multicast routers (i.e., run a multicast routing protocol) to ensure that the data for the host-joined multicast group is routed to the appropriate last-hop router and from there to the host.

36. In a group-shared tree, all senders send their multicast traffic using the same routing tree. With source-based tree, the multicast datagrams from a given source are routed over s specific routing tree constructed for that source; thus each source may have a different source-based tree and a router may have to keep track of several source-based trees for a given multicast group.

Chapter 4 Problems

Problem 1

a) With a connection-oriented network, every router failure will involve the routing of that connection. At a minimum, this will require the router that is "upstream" from the failed router to establish a new downstream part of the path to the destination node, with all of the requisite signaling involved in setting up a path. Moreover, all of the routers on the initial path that are downstream from the failed node must take down the failed connection, with all of the requisite signaling involved to do this.

With a connectionless datagram network, no signaling is required to either set up a new downstream path or take down the old downstream path. We have seen, however, that routing tables will need to be updated (e.g., either via a distance vector algorithm or a link state algorithm) to take the failed router into account. We have seen that with distance vector algorithms, this routing table change can sometimes be localized to the area near the failed router. Thus, a datagram network would be preferable. Interesting, the design criteria that the initial ARPAnet be able to function under stressful conditions was one of the reasons that datagram architecture was chosen for this Internet ancestor.

b) In order for a router to determine the delay (or a bound on delay) along an outgoing link, it would need to know the characteristics of the traffic from all sessions passing through that link. That is, the router must have per-session state in the router. This is possible in a connection-oriented network, but not with a connectionless network. Thus, a connection-oriented network would be preferable.

Problem 2

a) Maximum number of VCs over a link = 2^{16} = 65,536.
b) The centralized node could pick any VC number which is free from the set $\{0,1,\ldots,2^{16}-1\}$. In this manner, it is not possible that there are fewer VCs in progress than 65,536 without there being any common free VC number.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

c) Each of the links can independently allocate VC numbers from the set $\{0,1,\ldots,2^{16}-1\}$. Thus, a VC will likely have a different VC number for each link along its path. Each router in the VC's path must replace the VC number of each arriving packet with the VC number associated with the outbound link.

Problem 3

For a VC forwarding table, the columns are : Incoming Interface, Incoming VC Number, Outgoing Interface, Outgoing VC Number. For a datagram forwarding table, the columns are: Destination Address, Outgoing Interface.

Problem 4

c) No VC number can be assigned to the new VC; thus the new VC can be established in the network.
d) Each link has two available VC numbers. There are four links. So the number of combinations is $2^4 = 16$. One example combination is (10,00,00,10).

Problem 5

In a virtual circuit network, there *is* an end-to-end connection in the sense that each router along the path must maintain state for the connection; hence the terminology *connection service*. In a connection-oriented transport service over a connectionless network layer, such as TCP over IP, the end systems maintain connection state; however the routers have no notion of any connections; hence the terminology *connection-oriented service*.

Problem 6

To explain why there would be no input queuing, let's look at a specific design. For simplicity suppose each packet is the same size. We design the switch with time division multiplexing: time is broken into frames with each frame divided into n slots, with one slot needed to switch a packet through the fabric, and with one slot per frame devoted to each input line. Since at most one packet can arrive on each input line in each frame, the switching fabric will clear all packets in each frame.

Problem 7

a) No. To have no input port queuing, the switching fabric should operated at a speed at least 2 times the line speed.

b) No. To have no input port queuing, the switching fabric should operated at a speed at least 2 times the line speed.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 8

a)

Prefix Match	Link Interface
11100000	0
11100001 00000000	1
11100001	2
otherwise	3

b) Prefix match for first address is 4th entry: link interface 3
Prefix match for second address is 2nd entry: link interface 1
Prefix match for first address is 3rd entry: link interface 2

Problem 9

Destination Address Range	Link Interface
00000000 through 00111111	0
01000000 through 01111111	1
10000000 through 10111111	2
11000000 through 11111111	3

number of addresses in each range = $2^6 = 64$

Problem 10

Destination Address Range	Link Interface
10000000 through (64 addresses) 10111111	0
11000000 through(32 addresses) 11011111	1

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

11100000 through (32 addresses) 11111111	2
00000000 through (128 addresses) 01111111	3

Problem 11

223.1.17.0/25
223.1.17.128/26
223.1.17.192/26

Problem 12

Destination Address	**Link Interface**
200.23.16/21	0
200.23.24/24	1
200.23.24/21	2
otherwise	3

Problem 13

Destination Address	**Link Interface**
224/8	0
225.0/16	1
225/8	2
otherwise	3

Problem 14

Any IP address in range 101.101.101.65 to 101.101.101.127

Four equal size subnets: 101.101.101.64/28, 101.101.101.80/28, 101.101.101.96/28, 101.101.101.112/28

Problem 15

From 214.97.254/23, possible assignments are

a) Subnet A: 214.97.255/24 (256 addresses)
Subnet B: 214.97.254.0/25 - 214.97.254.0/29 (128-8 = 120 addresses)

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Subnet C: 214.97.254.128/25 (128 addresses)

Subnet D: 214.97.254.0/31 (2 addresses)
Subnet E: 214.97.254.2/31 (2 addresses)
Subnet F: 214.97.254.4/30 (4 addresses)

b) To simplify the solution, assume that no datagrams have router interfaces as ultimate destinations. Also, label D, E, F for the upper-right, bottom, and upper-left interior subnets, respectively.

Router 1

Longest Prefix Match	**Outgoing Interface**
11010110 01100001 11111111	Subnet A
11010110 01100001 11111110 0000000	Subnet D
11010110 01100001 11111110 000001	Subnet F

Router 2

Longest Prefix Match	**Outgoing Interface**
11010110 01100001 11111111 0000000	Subnet D
11010110 01100001 11111110 0	Subnet B
11010110 01100001 11111110 0000001	Subnet E

Router 3

Longest Prefix Match	**Outgoing Interface**
11010110 01100001 11111111 000001	Subnet F
11010110 01100001 11111110 0000001	Subnet E
11010110 01100001 11111110 1	Subnet C

Problem 16

The maximum size of data field in each fragment = 480 (20 bytes IP header). Thus the number of required fragments $= \left\lceil \frac{3000-20}{480} \right\rceil = 7$

Each fragment will have Identification number 422. Each fragment except the last one will be of size 500 bytes (including IP header). The last datagram will be of size 120 bytes (including IP header). The offsets of the 7 fragments will be 0, 60, 120, 180, 240, 300, 360. Each of the first 6 fragments will have flag=1; the last fragment will have flag=0.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 17

MP3 file size = 4 million bytes. Assume the data is carried in TCP segments, with each TCP segment also having 20 bytes of header. Then each datagram can carry 1500-40=1460 bytes of the MP3 file

Number of datagrams required = $= \left\lceil \frac{4 \times 10^6}{1460} \right\rceil = 2740$. All but the last datagram will be 1,500 bytes; the last datagram will be 1060+40 = 1100 bytes. Note that here there is not fragmentation – the source host does not create datagrams larger than 1500 bytes, and these datagrams are smaller than the MTUs of the links.

Problem 18

a) Home addresses: 192.168.0.1, 192.168.0.2, 192.168.0.3 with the router interface being 192.168.0.4

b)

NAT Translation Table

WAN Side	LAN Side
128.119.40.86, 4000	192.168.0.1, 3345
128.119.40.86, 4001	192.168.0.1, 3346
128.119.40.86, 4002	192.168.0.2, 3445
128.119.40.86, 4003	192.168.0.2, 3446
128.119.40.86, 4004	192.168.0.3, 3545
128.119.40.86, 4005	192.168.0.3, 3546

Problem 19

It is not possible to devise such a technique. In order to establish a direct TCP connection between Arnold and Bernard, either Arnold or Bob must initiate a connection to the other. But the NATs covering Arnold and Bob drop SYN packets arriving from the WAN side. Thus neither Arnold nor Bob can initiate a TCP connection to the other if they are both behind NATs.

Problem 20

v-w-y,
v-w-z-y, v-w-x-y, v-w-u-x-y, v-u-x-y, v-u-x-w-y, v-u-x-w-z-y, v-u-w-y, v-u-w-x-y, v-u-w-z-y, v-x-y, v-x-u-w-y, v-x-u-w-z-y, v-x-w-y, v-x-w-z-y

Problem 21

x to w:
x-w, x-v-w, x-v-u-w, x-u-v-w, x-u-w, x-y-w, x-y-z-w

w to u:
w-u, w-v-u, w-v-x-u, w-x-u, w-x-v-u, w-y-x-u, w-y-x-v-u, w-z-y-x-u, w-z-y-x-v-u

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

z to x:
z-y-x, z-y-w-x, z-y-w-u-x, z-y-w, z-y-w-v-x, z-y-w-u-v-x, z-y-w-v-u-x, z-w-u-x, z-w, z-w-v-x, z-w-u-v-x, z-w-v-u-x, z-w-y-x

Problem 22

Step	*N'*	*D(s),p(s)*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	X	∞	∞	∞	3,x	6,x	6,x	∞
1	xv	∞	7,v	6,v	3,x	6,x	4,v	∞
2	xvy	∞	7,v	6,v	3,x	6,x	4,v	18,y
3	xvyu	10,u	7,v	6,v	3,x	6,x	4,y	18,y
4	xvyuw	10,u	7,v	6,v	3,x	6,x	4,y	18,y
5	xvyuwt	8,t	7,v	6,v	3,x	6,x	4,y	12,t
6	xvyuwts	8,t	7,v	6,v	3,x	6,x	4,y	12,t
7	xvyuwtsz	8,t	7,v	6,v	3,x	6,x	4,y	12,t

Problem 23

a.

Step	*N'*	*D(x), p(x)*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	s	∞	1,s	4,s	∞	∞	∞	∞
1	st	∞	1,s	3,t	5,t	∞	8,t	6,t
2	stu	∞	1,s	3,t	5,t	6,u	8,t	6,t
3	stuv	8,v	1,s	3,t	5,t	6,u	6,v	6,t
4	stuvy	8,v	1,s	3,t	5,t	6,u	6,v	6,t
5	stuvyz	8,v	1,s	3,t	5,t	6,u	6,v	6,t
6	stuvyzw	8,v	1,s	3,t	5,t	6,u	6,v	6,t
7	stuvyzwx	8,v	1,s	3,t	5,t	6,u	6,v	6,t

b.

Step	*N'*	*D(x), p(x)*	*D(s),p(s)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	t	∞	1,t	2,t	4,t	∞	7,t	5,t
1	ts	∞	1,t	2,t	4,t	∞	7,t	5,t
2	tsu	∞	1,t	2,t	4,t	5,u	7,t	5,t
3	tsuv	7,v	1,t	2,t	4,t	5,u	5,v	5,t
4	tsuvw	7,v	1,t	2,t	4,t	5,u	5,v	5,t
5	tsuvwy	7,v	1,t	2,t	4,t	5,u	5,v	5,t
6	tsuvwyz	7,v	1,t	2,t	4,t	5,u	5,v	5,t
7	tsuvwyzx	7,v	1,t	2,t	4,t	5,u	5,v	5,t

c.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Step	*N'*	*D(x), p(x)*	*D(s),p(s)*	*D(t),p(t)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	u	∞	4,u	2,u	3,u	3,u	∞	∞
1	ut	∞	3,t	2,u	3,u	3,u	9,t	7,t
2	uts	∞	3,t	2,u	3,u	3,u	9,t	7,t
3	utsv	6,v	3,t	2,u	3,u	3,u	4,v	7,t
4	utsvw	6,v	3,t	2,u	3,u	3,u	4,v	7,t
5	utsvwy	6,v	3,t	2,u	3,u	3,u	4,v	7,t
6	utsvwyx	6,v	3,t	2,u	3,u	3,u	4,v	7,t
7	utsvwyxz	6,v	3,t	2,u	3,u	3,u	4,v	7,t

d.

Step	*N'*	*D(x), p(x)*	*D(s),p(s)*	*D(u),p(u)*	*D(t),pt)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	v	3,v	∞	3,v	4,v	4,v	1,v	∞
1	vy	3,v	∞	3,v	4,v	4,v	1,v	13,y
2	vyx	3,v	∞	3,v	4,v	4,v	1,v	13,y
3	vyxu	3,v	7,u	3,v	4,v	4,v	1,v	13,y
4	vyxut	3,v	5,t	3,v	4,v	4,v	1,v	9,t
5	vyxutw	3,v	5,t	3,v	4,v	4,v	1,v	9,t
6	vyxutws	3,v	5,t	3,v	4,v	4,v	1,v	9,t
7	vyxutwsz	3,v	5,t	3,v	4,v	4,v	1,v	9,t

e.

Step	*N'*	*D(x), p(x)*	*D(s),p(s)*	*D(u),p(u)*	*D(v),p(v)*	*D(t),p(t)*	*D(y),p(y)*	*D(z),p(z)*
0	w	6,w	∞	3,w	4,w	∞	∞	∞
1	wu	6,w	7,u	3,w	4,w	5,u	∞	∞
2	wuv	6,w	7,u	3,w	4,w	5,u	5,v	∞
3	wuvt	6,w	6,t	3,w	4,w	5,u	5,v	10,t
4	wuvty	6,w	6,t	3,w	4,w	5,u	5,v	10,t
5	wuvtys	6,w	6,t	3,w	4,w	5,u	5,v	10,t
6	wuvtysx	6,w	6,t	3,w	4,w	5,u	5,v	10,t
7	wuvtysxz	6,w	6,t	3,w	4,w	5,u	5,v	10,t

f.

Step	*N'*	*D(x), p(x)*	*D(s),p(s)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(t),p(t)*	*D(z),p(z)*
0	y	6,y	∞	∞	1,y	∞	7,y	12,y
1	yv	4,v	∞	4,t	1,y	5,t	5,v	12,y
2	yvx	4,v	∞	4,t	1,y	5,t	5,v	12,y
3	yvxu	4,v	8,u	4,t	1,y	5,t	5,v	12,y
4	yvxuw	4,v	8,u	4,t	1,y	5,t	5,v	12,y
5	yvxuwt	4,v	6,t	4,t	1,y	5,t	5,v	10,t

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

6	yvxuwts	4,v	6,t	4,t	1,y	5,t	5,v	10,t
7	yvxuwtsz	4,v	6,t	4,t	1,y	5,t	5,v	10,t

g.

Step	*N'*	*D(x), p(x)*	*D(s),p(s)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(t),p(t)*
0	z	∞	∞	∞	∞	∞	12,z	5,z
1	zt	∞	6,t	7,t	9,t	∞	12,z	5,z
2	zts	∞	6,t	7,t	9,t	∞	12,z	5,z
3	ztsu	∞	6,t	7,t	9,t	10,u	12,z	5,z
4	ztsuv	12,v	6,t	7,t	9,t	10,u	10,v	5,z
5	ztsuvw	12,v	6,t	7,t	9,t	10,u	10,v	5,z
6	ztsuvwy	12,v	6,t	7,t	9,t	10,u	10,v	5,z
7	ztsuvwyx	12,v	6,t	7,t	9,t	10,u	10,v	5,z

Problem 24

Cost to

		u	v	x	y	z
	v	∞	∞	∞	∞	∞
From	x	∞	∞	∞	∞	∞
	y	∞	∞	∞	∞	∞
	z	∞	5	2	10	0

Cost to

		u	v	x	y	z
	v	4	0	∞	7	5
From	x	12	∞	0	1	2
	y	∞	7	1	0	10
	z	9	5	2	3	0

Cost to

		u	v	x	y	z
	v	4	0	7	7	5
From	x	12	7	0	1	2
	y	11	7	1	0	3
	z	9	5	2	3	0

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

		Cost to				
		u	v	x	y	z
	v	4	0	7	7	5
From	x	11	7	0	1	2
	y	11	7	1	0	3
	z	9	5	2	3	0

Problem 25

The wording of this question was a bit ambiguous. We meant this to mean, "the number of iterations from when the algorithm is run for the first time" (that is, assuming the only information the nodes initially have is the cost to their nearest neighbors). We assume that the algorithm runs synchronously (that is, in one step, all nodes compute their distance tables at the same time and then exchange tables).

At each iteration, a node exchanges distance tables with its neighbors. Thus, if you are node A, and your neighbor is B, all of B's neighbors (which will all be one or two hops from you) will know the shortest cost path of one or two hops to you after one iteration (i.e., after B tells them its cost to you).

Let d be the "diameter" of the network - the length of the longest path without loops between any two nodes in the network. Using the reasoning above, after $d-1$ iterations, all nodes will know the shortest path cost of d or fewer hops to all other nodes. Since any path with greater than d hops will have loops (and thus have a greater cost than that path with the loops removed), the algorithm will converge in at most $d-1$ iterations.

ASIDE: if the DV algorithm is run as a result of a change in link costs, there is no a priori bound on the number of iterations required until convergence unless one also specifies a bound on link costs.

Problem 26

a. $D_x(y) = 5$, $D_x(w) = 2$, $D_x(u) = 7$

b. First consider what happens if c(x,y) changes. If c(x,y) becomes larger or smaller (as long as $c(x,y) > 0$), the least cost path from x to u will still have cost at least 7. Thus a change in c(x,y) will not cause x to inform its neighbors of any changes. Now consider if c(x,w) changes. If $c(x,w) = \varepsilon \leq 1$, then the least-cost path to u continues to pass through w and its cost changes to $5 + \varepsilon$; x will inform its neighbors of this new cost. If $c(x,w) = \delta > 6$, then the least cost path now passes through y and has cost 11; again x will inform its neighbors of this new cost.

c. Any change in link cost c(x,y) will not cause x to inform its neighbors of a new minimum-cost path to u .

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 27

Node x table

		Cost to		
		x	y	z
	x	0	5	2
From	y	∞	∞	∞
	z	∞	∞	∞

		Cost to		
		x	y	z
	x	0	5	2
From	y	5	0	6
	z	2	6	0

Node y table

		Cost to		
		x	y	z
	x	∞	∞	∞
From	y	5	0	6
	z	∞	∞	∞

		Cost to		
		x	y	z
	x	0	5	2
From	y	5	0	6
	z	2	6	0

Node z table

		Cost to		
		x	y	z
	x	∞	∞	∞
From	y	∞	∞	∞
	z	2	6	0

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

		Cost to		
		x	y	z
	x	0	5	2
From	y	5	0	6
	z	2	6	0

Problem 28

Since full AS path information is available from an AS to a destination in BGP, loop detection is simple – if a BGP peer receives a route that contains its own AS number in the AS path, then using that route would result in a loop.

Problem 29

a. eBGP

b. iBGP

c. eBGP

d. iBGP

Problem 30

a) l_1 because this interface begins the least cost path from 1d towards the gateway router 1c.

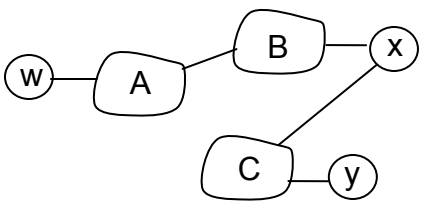
b) l_2. Both routes have equal AS-PATH length but l_2 begins the path that has the closest NEXT-HOP router.

c) l_1. l_1 begins the path that has the shortest AS-PATH.

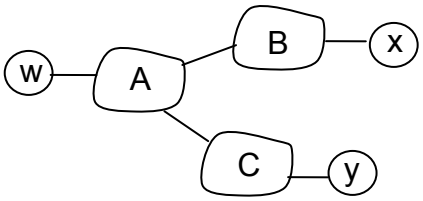
Problem 31

One way for C to force B to hand over all of B's traffic to D on the east coast is for C to only advertise its route to D via its east coast peering point with C.

Problem 32



X's view of the topology



W's view of the topology

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

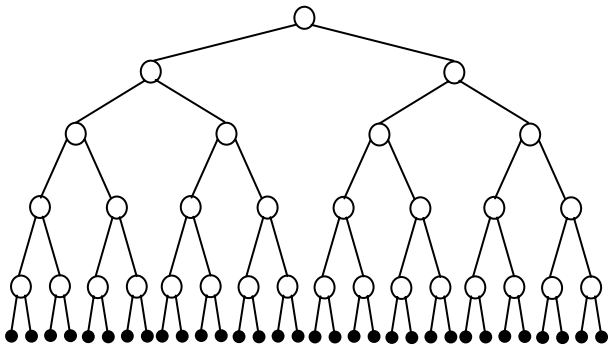
In the above solution, X does not know about the AC link since X does not receive an advertised route to w or to y that contain the AC link (i.e., X receives no advertisement containing both AS A and AS C on the path to a destination.

Problem 33

The minimal spanning tree has s connected to t at a cost of 1, u connected to t at a cost of 2; v connected to u at a cost of 1; w connected to v at cost of 1; y connected to v at a cost of 1.

We can argue informally that the tree cost is minimal as follows. u, v, w, and y can be connected to each other at a cost of 3. One can't do any better than that. In order to connect s in to any of u, v, w, y the minimum cost is 3 (via t).

Problem 34



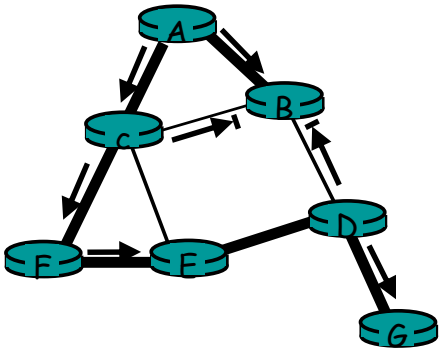
The 32 receives are shown connected to the sender in the binary tree configuration shown above. With network-layer broadcast, a copy of the message is forwarded over each link exactly once. There are thus 62 link crossings (2+4+8+16+32). With unicast emulation, the sender unicasts a copy to each receiver over a path with5 hops. There are thus 160 link crossings (5*32).

A topology in which all receivers are in a line, with the sender at one send of the line, will have the largest disparity between the cost of network-layer broadcast and unicast emulation.

Problem 35

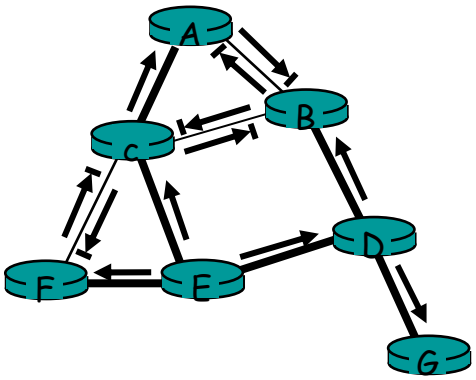
This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)



The thicker shaded lines represent The shortest path tree from A to all destination. Other solutions are possible, but in these solutions, B can not route to either C or D from

Problem 36



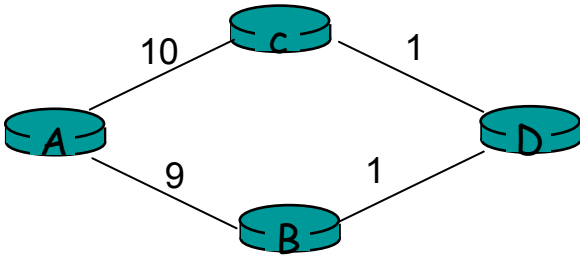
Problem 37

The center-based tree for the topology shown in the original figure connects A to D; B to C; E to C; and F to C (all directly). This center-based tree is different from the minimal spanning tree shown in the figure.

Problem 38

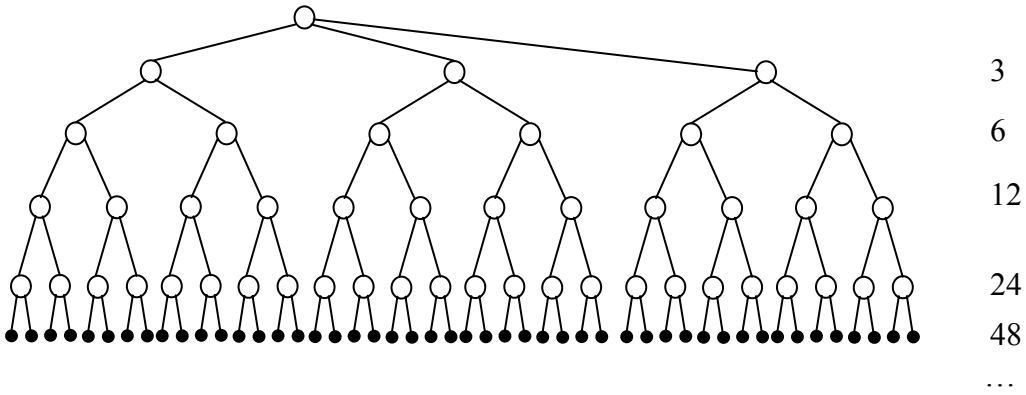
Dijkstra's algorithm for the network below, with node A as the source, results in a least-unicast-cost path tree of links AC, AB, and BD, with an overall free cost of 20. The minimum spanning tree contains links AB, BD, and DC, at a cost of 11.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)



Problem 39

After 1 step 3 copies are transmitted, after 2 steps 6 copies are transmitted. After 3 steps, 12 copies are transmitted, and so on. After k steps, $3*2^{k-1}$ copies will be transmitted in that step.



Problem 40

The protocol must be built at the application layer. For example, an application may periodically multicast its identity to all other group members in an application-layer message.

Problem 41

A simple application-layer protocol that will allow all members to know the identity of all other members in the group is for each instance of the application to send a multicast message containing its identity to all other members. This protocol sends message in-band, since the multicast channel is used to distribute the identification messages as well as multicast data from the application itself. The use of the in-band signaling makes use of the existing multicast distribution mechanism, leading to a very simple design.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 42

$32 - 4 = 28$ bits are available for multicast addresses. Thus, the size of the multicast address space is $N = 2^{28}$.

The probability that two groups choose the same address is

$$\frac{1}{N} = 2^{-28} = 3.73 \cdot 10^{-9}$$

The probability that 1000 groups all have different addresses is

$$\frac{N \cdot (N-1) \cdot (N-2) \Lambda \ (N-999)}{N^{1000}} = \left(1 - \frac{1}{N}\right)\left(1 - \frac{2}{N}\right) \Lambda \ \left(1 - \frac{999}{N}\right)$$

Ignoring cross-product terms, this is approximately equal to

$$1 - \left(\frac{1 + 2 + \Lambda \ + 999}{N}\right) = 1 - \frac{999 \cdot 1000}{2N} = 0.998$$

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 5 Review Questions

1. Although each link guarantees that an IP datagram sent over the link will be received at the other end of the link without errors, it is not guaranteed that IP datagrams will arrive at the ultimate destination in the proper order. With IP, datagrams in the same TCP connection can take different routes in the network, and therefore arrive out of order. TCP is still needed to provide the receiving end of the application the byte stream in the correct order. Also, IP can lose packets due to routing loops or equipment failures.

2. Framing: there is also framing in IP and TCP; link access; reliable delivery: there is also reliable delivery in TCP; flow control: there is also flow control in TCP; error detection: there is also error detection in IP and TCP; error correction; full duplex: TCP is also full duplex.

3. There will be a collision in the sense that while a node is transmitting it will start to receive a packet from the other node.

4. Slotted Aloha: 1, 2 and 4 (slotted ALOHA is only partially decentralized, since it requires the clocks in all nodes to be synchronized). Token ring: 1, 2, 3, 4.

5. In polling, a discussion leader allows only one participant to talk at a time, with each participant getting a chance to talk in a round-robin fashion. For token ring, there isn't a discussion leader, but there is wine glass that the participants take turns holding. A participant is only allowed to talk if the participant is holding the wine glass.

6. When a node transmits a frame, the node has to wait for the frame to propagate around the entire ring before the node can release the token. Thus, if L/R is small as compared to t_{prop}, then the protocol will be inefficient.

7. 2^{48} MAC addresses; 2^{32} IPv4 addresses; 2^{128} IPv6 addresses.

8. C's adapter will process the frames, but the adapter will not pass the datagrams up the protocol stack. If the LAN broadcast address is used, then C's adapter will both process the frames and pass the datagrams up the protocol stack.

9. An ARP query is sent in a broadcast frame because the querying host does not which adapter address corresponds to the IP address in question. For the response, the sending node knows the adapter address to which the response should be sent, so there is no need to send a broadcast frame (which would have to be processed by all the other nodes on the LAN).

10. No it is not possible. Each LAN has its own distinct set of adapters attached to it, with each adapter having a unique LAN address.

11. The three Ethernet technologies have identical frame structures.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

12. 20 million transitions per second.

13. After the 5th collision, the adapter chooses from {0, 1, 2,…, 31}. The probability that it chooses 4 is 1/32. It waits 204.8 microseconds.

14. 2 (the internal subnet and the external internet)

Chapter 5 Problems

Problem 1

The rightmost column and bottom row are for parity bits.

1 0 1 0 0
1 0 1 0 0
1 0 1 0 0
1 0 1 1 1
0 0 0 1 1

Problem 2

Suppose we begin with the initial two-dimensional parity matrix:

0 0 0 0
1 1 1 1
0 1 0 1
1 0 1 0

With a bit error in row 2, column 3, the parity of row 2 and column 3 is now wrong in the matrix below:

0 0 0 0
1 1 0 1
0 1 0 1
1 0 1 0

Now suppose there is a bit error in row 2, column 2 and column 3. The parity of row 2 is now correct! The parity of columns 2 and 3 is wrong, but we can't detect in which rows the error occurred!

0 0 0 0
1 0 0 1
0 1 0 1
1 0 1 0

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

The above example shows that a double bit error can be detected (if not corrected).

Problem 3

To compute the Internet checksum, we add up the values at 16-bit quantities:

```
00000000  00000001
00000010  00000011
00000100  00000101
00000110  00000111
00001000  00001001
-------------------------
00010100  00011001
```

The one's complement of the sum is 11101011 11100110.

Problem 4

a) To compute the Internet checksum, we add up the values at 16-bit quantities:

```
00000001  00000010
00000011  00000100
00000101  00000110
00000111  00001000
00001001  00001010
-------------------------
00011001  00011110
```

The one's complement of the sum is 11100110 11100001.

b) To compute the Internet checksum, we add up the values at 16-bit quantities:

```
01000001  01000010
01000011  01000100
01000101  01000110
01000111  01001000
01001001  01001010
-------------------------
01011000  01011111
```

The one's complement of the sum is 10100111 10100000

c) To compute the Internet checksum, we add up the values at 16-bit quantities:

```
01100001  01100010
```

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

01100011 01100100
01100101 01100110
01100111 01100111
01101000 01101001

11111001 11111101
The one's complement of the sum is 00000110 00000010.

Problem 5

If we divide 1001 into 10101010000 we get 10111101, with a remainder of R = 101.

Problem 6

a) If we divide 1001 into 10010001000 we get 10000001, with a remainder of R = 001.
b) If we divide 1001 into we get 10100011000 we get 10110101, with a remainder of R = 101.
c) If we divide 1001 into 010101010000 we get 010111101, with a remainder of R = 101.

Problem 7

a)

$$\begin{aligned} E(p) &= Np(1-p)^{N-1} \\ E'(p) &= N(1-p)^{N-1} - Np(N-1)(1-p)^{N-2} \\ &= N(1-p)^{N-2}((1-p) - p(N-1)) \end{aligned}$$

$$E'(p) = 0 \Rightarrow p^* = \frac{1}{N}$$

b)

$$E(p^*) = N\frac{1}{N}(1-\frac{1}{N})^{N-1} = (1-\frac{1}{N})^{N-1} = \frac{(1-\frac{1}{N})^N}{1-\frac{1}{N}}$$

$$\lim_{N\to\infty}(1-\frac{1}{N}) = 1 \qquad \lim_{N\to\infty}(1-\frac{1}{N})^N = \frac{1}{e}$$

Thus

$$\lim_{N\to\infty} E(p^*) = \frac{1}{e}$$

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 8

$$E(p) = Np(1-p)^{2(N-1)}$$
$$E'(p) = N(1-p)^{2(N-2)} - Np2(N-1)(1-p)^{2(N-3)}$$
$$= N(1-p)^{2(N-3)}((1-p) - p2(N-1))$$

$$E'(p) = 0 \Rightarrow p^* = \frac{1}{2N-1}$$

$$E(p^*) = \frac{N}{2N-1}(1 - \frac{1}{2N-1})^{2(N-1)}$$

$$\lim_{N\to\infty} E(p^*) = \frac{1}{2} \cdot \frac{1}{e} = \frac{1}{2e}$$

Problem 9

a) $(1 - p(a))^3\, p(A)$
where,
$p(A)$ = probability that A succeeds in a slot

$p(A) = p(A$ *transmits and B does not and C does not)*
$= p(A$ *transmits)* $p(B$ *does not transmit)* $p(C$ *does not transmit)*
$= p(1 - p)\,(1 - p) = p(1 - p)^2$

Hence, *p(A succeeds for first time in slot 4)* $= (1 - p(a))^3\, p(A) = (1 - p(1 - p)^2)^3\, p(1 - p)^2$

b) *p(A succeeds in slot 2)* $= p(1-p)^2$
p(B succeeds in slot 2) $= p(1-p)^2$
p(C succeeds in slot 2) $= p(1-p)^2$

p(either A or B or C succeeds in slot 2) $= 3\, p(1-p)^2$
(because these events are mutually exclusive)

c) *p(some node succeeds in a slot)* $= 3\, p(1-p)^2$
p(no node succeeds in a slot) $= 1 - 3\, p(1-p)^2$

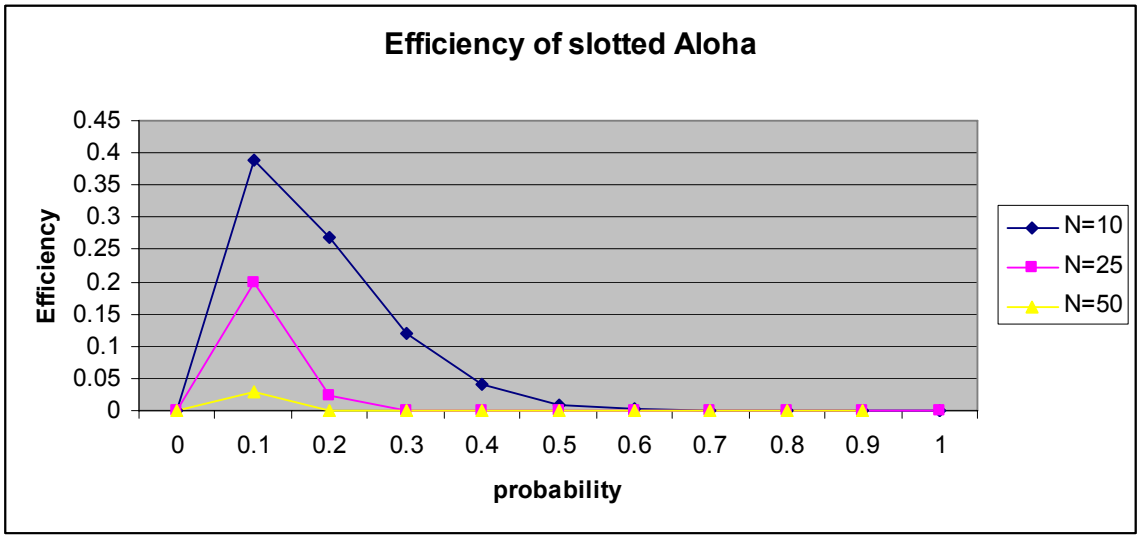
Hence, *p(first success occurs in slot 4) = p(no node succeeds in first 3 slots) p(some node succeeds in* 4th *slot)* $= (1 - 3\, p(1-p)^2)^3\; 3\, p(1-p)^2$

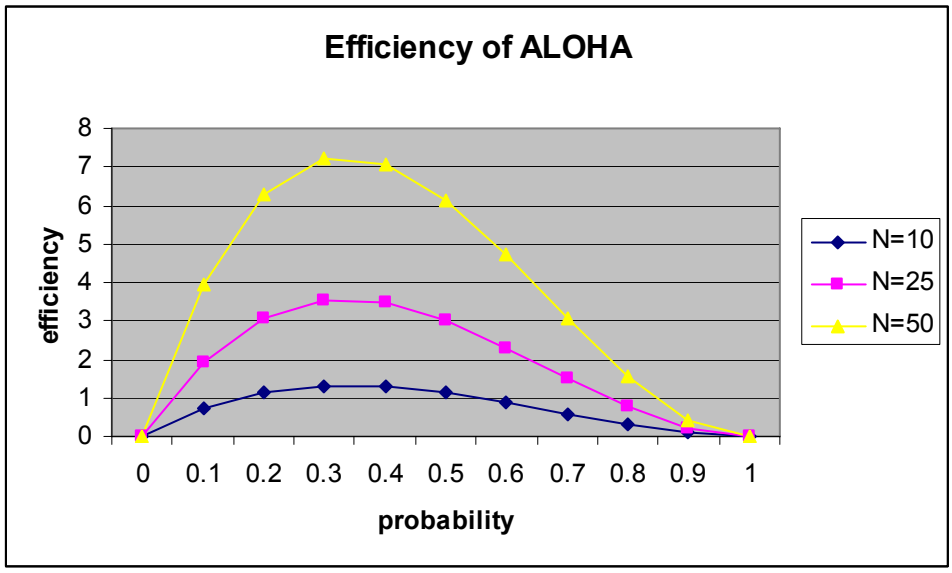
d) efficiency = *p(success in a slot)* $= 3\, p(1-p)^2$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 10





Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 11

The length of a polling round is

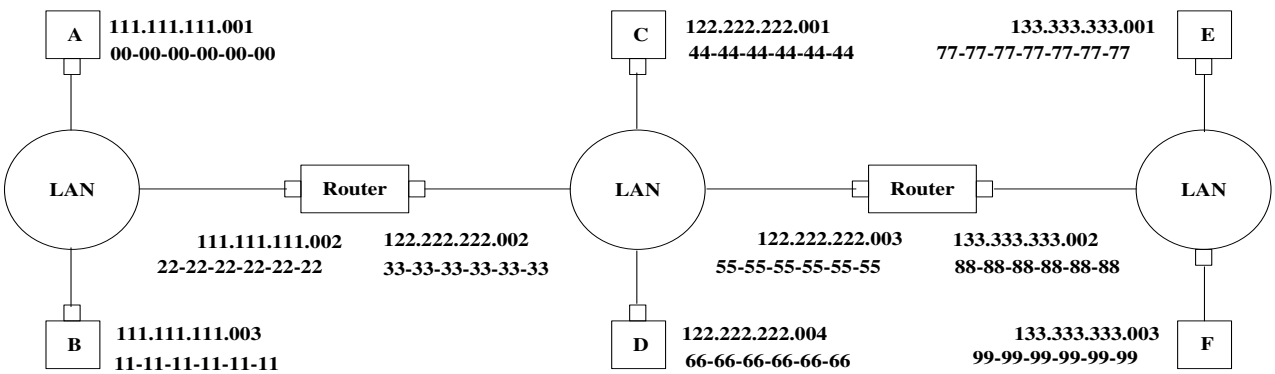
$$N(Q/R + d_{poll}).$$

The number of bits transmitted in a polling round is NQ. The maximum throughput therefore is

$$\frac{NQ}{N(Q/R + d_{poll})} = \frac{R}{1 + \frac{d_{poll}R}{Q}}$$

Problem 12

a), b), c) See figure below.



d)

1. Forwarding table in A determines that the datagram should be routed to interface 111.111.111.002.
2. The adapter in A creates and Ethernet packet with Ethernet destination address 22-22-22-22-22-22.
3. The first router receives the packet and extracts the datagram. The forwarding table in this router indicates that the datagram is to be routed to 122.222.222.003.
4. The first router then sends the Ethernet packet with the destination address of 55-55-55-55-55-55 and source address of 33-33-33-33-33-33 via its interface with IP address of 122.222.222.002.
5. The process continues until the packet has reached Host F.

e)

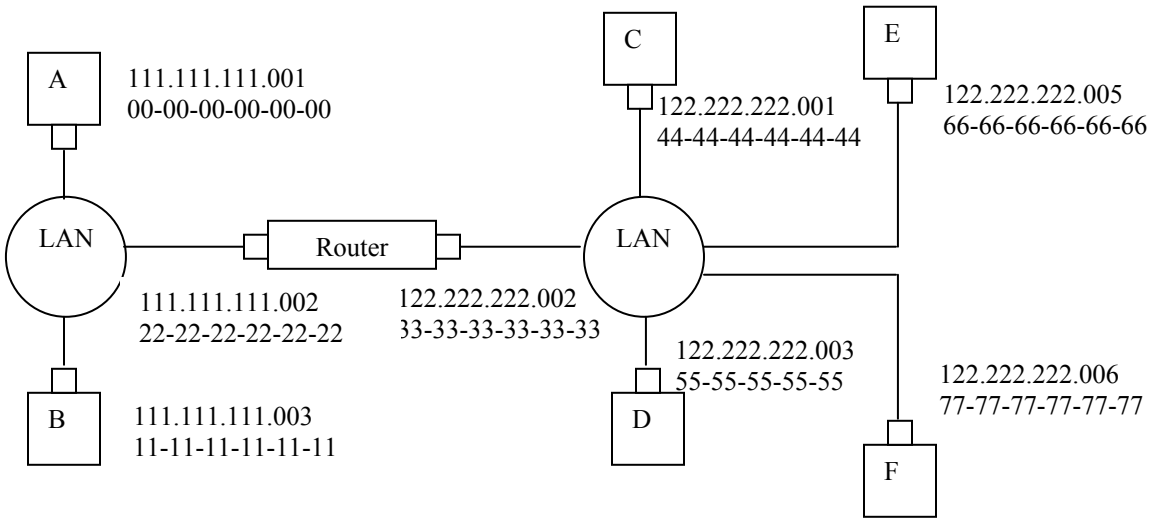
ARP in A must now determine the LAN address of 111.111.111.002. Host A sends out an ARP query packet within a broadcast Ethernet frame. The first router receives the query packet and sends to Host A an ARP response packet. This ARP response packet is carried by an Ethernet frame with Ethernet destination address 00-00-00-00-00-00.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 13

a) b) c) See figure below:



d)

1. Forwarding table in A determines that the datagram should be routed to interface 111.111.111.002.
2. The adapter in A creates and Ethernet packet with Ethernet destination address 22-22-22-22-22-22.
3. The router receives the packet and extracts the datagram. The forwarding table in this router indicates that the datagram is to be routed via interface 122.222.222.002.
4. The router then replaces the destination MAC address in the datagram it received from host A, with the MAC address of F (77-77-77-77-77-77) and sends the datagram over the interface 122.222.222.002.
5. The switch (that is connected to the interface 122.222.222.002 of the router) forwards the datagram to its interface where F is connected. The datagram then reaches F.

e) ARP in A must now determine the MAC address of 111.111.111.002. Host A sends out an ARP query packet within a broadcast Ethernet frame. The router receives the query packet and sends to Host A an ARP response packet. This ARP response packet is carried by an Ethernet frame with Ethernet destination address 00-00-00-00-00-00.

Problem 14

Wait for 51,200 bit times. For 10 Mbps, this wait is

$$\frac{51.2 \times 10^3 \, bits}{10 \times 10^6 \, bps} = 5.12 \, \text{msec}$$

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

For 100 Mbps, the wait is 512 μ sec.

Problem 15

At $t = 0$ A transmits. At $t = 576$, A would finish transmitting. In the worst case, B begins transmitting at time $t = 224$. At time $t = 224 + 225 = 449$ B's first bit arrives at A. Because $449 < 576$, A aborts before completing the transmission of the packet, as it is supposed to do.

Thus A cannot finish transmitting before it detects that B transmitted. This implies that if A does not detect the presence of a host, then no other host begins transmitting while A is transmitting.

Problem 16

Time, t	Event
0	A and B begin transmission
225	A and B detect collision
273	A and B finish transmitting jam signal
273+225 = 498	B's last bit arrives at A; A detects an idle channel
498+96=594	A starts transmitting
273+512 = 785	B returns to Step2 B must sense idle channel for 96 bit times before it transmits
594+225=819	A's transmission reaches B

Because A's retransmission reaches B before B's scheduled retransmission time, B refrains from transmitting while A retransmits. Thus A and B do not collide. Thus the factor 512 appearing in the exponential backoff algorithm is sufficiently large.

Problem 17

We want $1/(1+5a) = .5$ or, equivalently, $a = .2 = t_{prop} / t_{trans}$. $t_{prop} = d/(1.8 \times 10^8)$ m/sec and $t_{trans} = (576 \text{ bits})/(10^8 \text{ bits/sec}) = 5.76\mu$ sec. Solving for d we obtain $d = 265$ meters. For the 100 Mbps Ethernet standard, the maximum distance between two hosts is 200 m.

For transmitting station A to detect whether any other station transmitted during A's interval, t_{trans} must be greater than $2t_{prop} = 2 \cdot 265\text{m}/1.8 \times 10^8 \text{ m/sec} = 2.94\mu$ sec. Because $2.94 < 5.76$, A will detect B's signal before the end of its transmission.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 18

a)
Let Y be a random variable denoting the number of slots until a success:

$$P(Y = m) = \beta(1-\beta)^{m-1},$$

where β is the probability of a success.

This is a geometric distribution, which has mean $1/\beta$. The number of consecutive wasted slots is $X = Y - 1$ that

$$x = E[X] = E[Y] - 1 = \frac{1-\beta}{\beta}$$

$$\beta = Np(1-p)^{N-1}$$

$$x = \frac{1 - Np(1-p)^{N-1}}{Np(1-p)^{N-1}}$$

$$\text{efficiency} = \frac{k}{k+x} = \frac{k}{k + \dfrac{1 - Np(1-p)^{N-1}}{Np(1-p)^{N-1}}}$$

b)

Maximizing efficiency is equivalent to minimizing x, which is equivalent to maximizing β. We know from the text that β is maximized at $p = \dfrac{1}{N}$.

c)

$$\text{efficiency} = \frac{k}{k + \dfrac{1-(1-\frac{1}{N})^{N-1}}{(1-\frac{1}{N})^{N-1}}}$$

$$\lim_{N\to\infty} \text{efficiency} = \frac{k}{k + \dfrac{1-1/e}{1/e}} = \frac{k}{k+e-1}$$

d) Clearly, $\dfrac{k}{k+e-1}$ approaches 1 as $k \to \infty$.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 19

a)

$$\frac{900m}{2\cdot 10^8 m/\sec} + 4\cdot\frac{20bits}{10\times 10^6 bps}$$

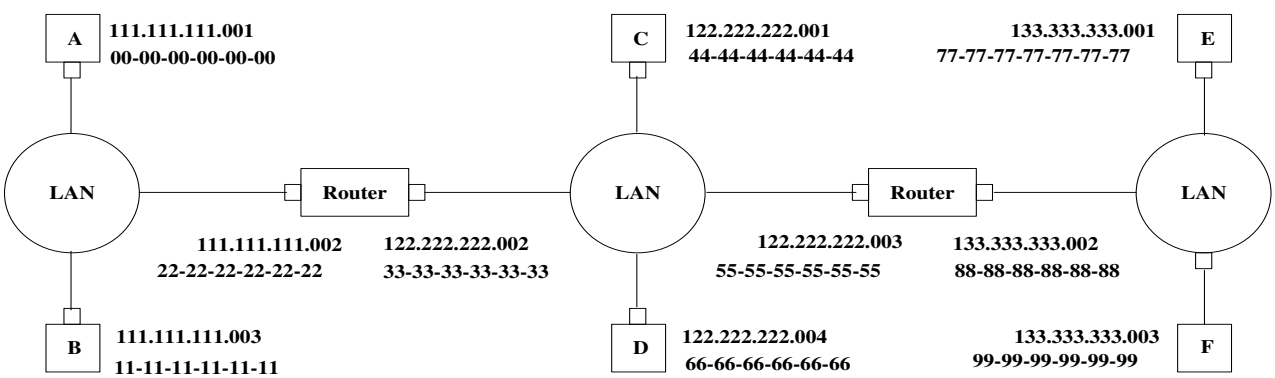
$$= (4.5\times 10^{-6} + 8\times 10^{-6})\sec$$
$$= 12.5\mu\sec$$

b)

- At time $t = 0$, both A and B transmit.
- At time $t = 12.5\mu\sec$, A detects a collision.
- At time $t = 25\mu\sec$ last bit of B's aborted transmission arrives at A.
- At time $t = 37.5\mu\sec$ first bit of A's retransmission arrives at B.
- At time $t = 37.5\mu\sec + \frac{1000bits}{10\times 10^6 bps} = 137.5\mu\sec$ A's packet is completely delivered at B.

c) $12.5\mu\sec + 5\cdot 100\mu\sec = 512.5\mu\sec$

Problem 20



i) from A to left router: Source MAC address: 00-00-00-00-00-00
Destination MAC address: 22-22-22-22-22-22
Source IP: 111.111.111.001
Destination IP: 133.333.333.003

ii) from the left router to the right router: Source MAC address: 33-33-33-33-33-33
Destination MAC address: 55-55-55-55-55-55
Source IP: 111.111.111.001
Destination IP: 133.333.333.003

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

iii) from the right router to F: Source MAC address: 88-88-88-88-88-88
Destination MAC address: 99-99-99-99-99-99
Source IP: 111.111.111.001
Destination IP: 133.333.333.003

Problem 21

i) from A to switch: Source MAC address: 00-00-00-00-00-00
Destination MAC address: 55-55-55-55-55-55

Source IP: 111.111.111.001
Destination IP: 133.333.333.003

ii) from switch to right router: Source MAC address: 00-00-00-00-00-00
Destination MAC address: 55-55-55-55-55-55

Source IP: 111.111.111.001
Destination IP: 133.333.333.003

iii) from right router to F: Source MAC address: 88-88-88-88-88-88
Destination MAC address: 99-99-99-99-99-99

Source IP: 111.111.111.001
Destination IP: 133.333.333.003

Problem 22

If all the 14 nodes send out data at the maximum possible rate of 100 Mbps, a total aggregate throughput of 14*100 = 1400 Mbps is possible.

Problem 23

Each departmental hub is a single collision domain that can have a maximum throughput of 100 Mbps. The links connecting the web server and the mail server has a maximum throughput of 100 Mbps. Hence, if the three collision domains and the web server and mail server send out data at their maximum possible rates of 100 Mbps each, a maximum total aggregate throughput of 500 Mbps can be achieved among the 14 end systems.

Problem 24

All of the 14 end systems will lie in the same collision domain. In this case, the maximum total aggregate throughput of 100 Mbps is possible among the 14 end sytems.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 25

Action	Switch Table State	Link(s) packet is forwarded to	Explanation
A sends a frame to D	Switch learns interface corresponding to MAC address of A	B, C, D, E, and F	Since switch table is empty, so switch does not know the interface corresponding to MAC address of D
D replies with a frame to A	Switch learns interface corresponding to MAC address of D	A	Since switch already knows interface corresponding to MAC address of A
C sends a frame to D	Switch learns the interface corresponding to MAC address of C	D	Since switch already knows the interface corresponding to MAC address of D
D replies with a frame to C	Switch table state remains the same as before	C	Since switch already knows the interface corresponding to MAC address of C

Problem 26

The time required to fill $L \cdot 8$ bits is

$$\frac{L \cdot 8}{64 \times 10^3} \sec = \frac{L}{8} m \sec .$$

b) For $L = 1{,}500,$ the packetization delay is

$$\frac{1500}{8} m \sec = 187.5 m \sec .$$

For $L = 48,$ the packetization delay is

$$\frac{48}{8} m \sec = 6 m \sec .$$

c)

$$\text{Store-and-forward delay } = \frac{L \cdot 8 + 40}{R}$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

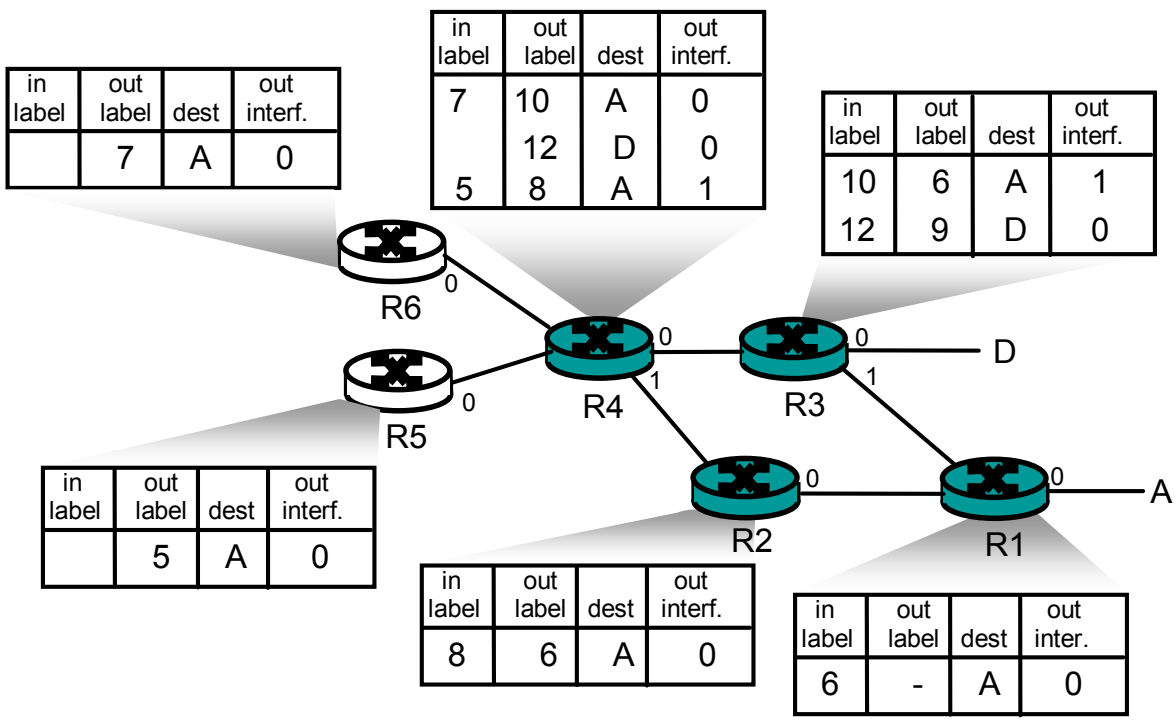
For $L = 1{,}500$, the delay is

$$\frac{1500 \cdot 8 + 40}{155 \times 10^6} \text{sec} \approx \frac{12}{155} m\,\text{sec} \approx 77 \mu\,\text{sec}$$

For $L = 48$, store-and-forward delay $< 1 \mu\,\text{sec}$.

d) Store-and-forward delay is small for both cases for typical ATM link speeds. However, packetization delay for $L = 1500$ is too large for real-time voice applications.

Problem 27



in label	out label	dest	out interf.
	7	A	0

in label	out label	dest	out interf.
7	10	A	0
	12	D	0
5	8	A	1

in label	out label	dest	out interf.
10	6	A	1
12	9	D	0

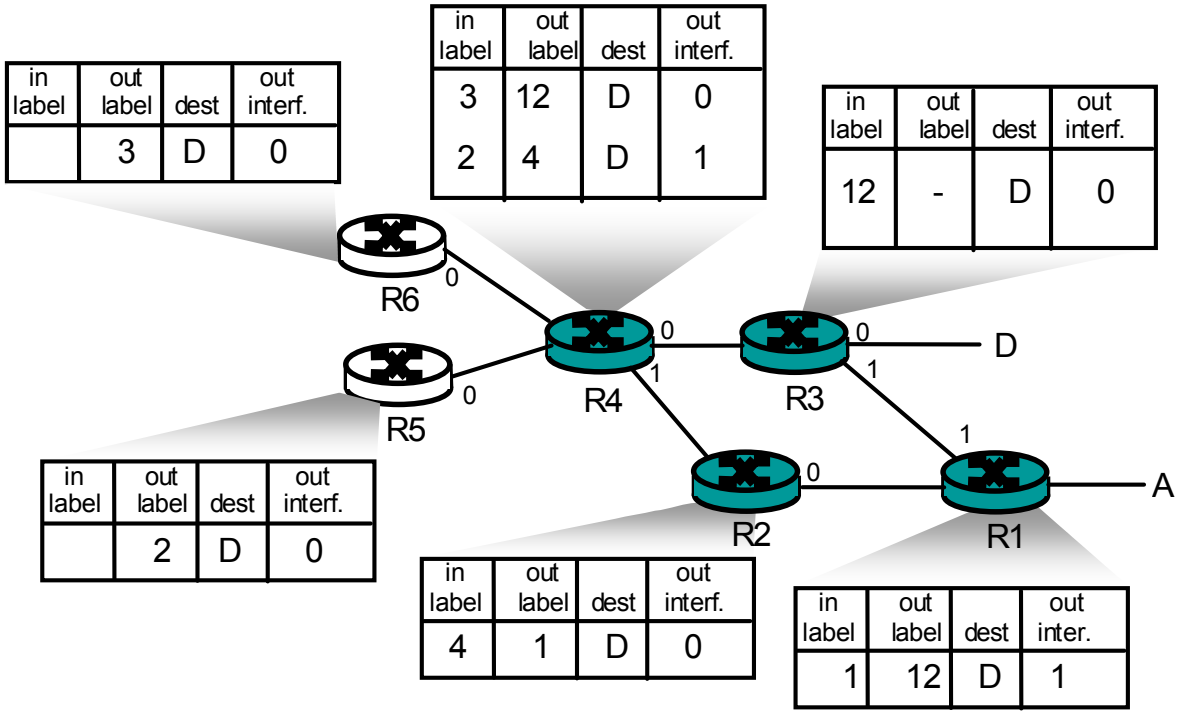
in label	out label	dest	out interf.
	5	A	0

in label	out label	dest	out interf.
8	6	A	0

in label	out label	dest	out inter.
6	-	A	0

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 28



in label	out label	dest	out interf.
	3	D	0

in label	out label	dest	out interf.
3	12	D	0
2	4	D	1

in label	out label	dest	out interf.
12	-	D	0

in label	out label	dest	out interf.
	2	D	0

in label	out label	dest	out interf.
4	1	D	0

in label	out label	dest	out inter.
1	12	D	1

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 6 Review Questions

1. In infrastructure mode of operation, each wireless host is connected to the larger network via a base station (access point). If not operating in infrastructure mode, a network operates in ad-hoc mode. In ad-hoc mode, wireless hosts have no infrastructure with which to connect. In the absence of such infrastructure, the hosts themselves must provide for services such as routing, address assignment, DNS-like name translation, and more.

2. a) Single hop, infrastructure-based
   b) Single hop, infrastructure-less
   c) Multi-hop, infrastructure-based
   d) Multi-hop, infrastructure-less

3. Path loss is due to the attenuation of the electromagnetic signal when it travels through matter. Multipath propagation results in blurring of the received signal at the receiver and occurs when portions of the electromagnetic wave reflect off objects and ground, taking paths of different lengths between a sender and receiver. Interference from other sources occurs when the other source is also transmitting in the same frequency range as the wireless network.

4. a) Increasing the transmission power
   b) Reducing the transmission rate

5. APs transmit beacon frames. An AP's beacon frames will be transmitted over one of the 11 channels. The beacon frames permit nearby wireless stations to discover and identify the AP.

6. False

7. APs transmit beacon frames. An AP's beacon frames will be transmitted over one of the 11 channels. The beacon frames permit nearby wireless stations to discover and identify the AP.

8. False

9. Each wireless station can set an RTS threshold such that the RTS/CTS sequence is used only when the data frame to be transmitted is longer than the threshold. This ensures that RTS/CTS mechanism is used only for large frames.

10. No, there wouldn't be any advantage. Suppose there are two stations that want to transmit at the same time, and they both use RTS/CTS. If the RTS frame is as long as a DATA frames, the channel would be wasted for as long as it would have been wasted for two colliding DATA frames. Thus, the RTS/CTS exchange is only useful when the RTS/CTS frames are significantly smaller than the DATA frames.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

11. Initially the switch has an entry in its forwarding table which associates the wireless station with the earlier AP. When the wireless station associates with the new AP, the new AP creates a frame with the wireless station's MAC address and broadcasts the frame. The frame is received by the switch. This forces the switch to update its forwarding table, so that frames destined to the wireless station are sent via the new AP.

12. Any ordinary Bluetooth node can be a master node whereas access points in 802.11 networks are special devices (normal wireless devices like laptops cannot be used as access points).

13. False

14. "Opportunistic Scheduling" refers to matching the physical layer protocol to channel conditions between the sender and the receiver, and choosing the receivers to which packets will be sent based on channel condition. This allows the base station to make best use of the wireless medium.

15. UMTS to GSM and CDMA-2000 to IS-95.

16. No. A node can remain connected to the same access point throughout its connection to the Internet (hence, not be mobile). A mobile node is the one that changes its point of attachment into the network over time. Since the user is always accessing the Internet through the same access point, she is not mobile.

17. A permanent address for a mobile node is its IP address when it is at its home network. A care-of-address is the one its gets when it is visiting a foreign network. The COA is assigned by the foreign agent (which can be the edge router in the foreign network or the mobile node itself).

18. False

19. The home network in GSM maintains a database called the home location register (HLR), which contains the permanent cell phone number and subscriber profile information about each of its subscribers. The HLR also contains information about the current locations of these subscribers. The visited network maintains a database known as the visitor location register (VLR) that contains an entry for each mobile user that is currently in the portion of the network served by the VLR. VLR entries thus come and go as mobile users enter and leave the network.

    The edge router in home network in mobile IP is similar to the HLR in GSM and the edge router in foreign network is similar to the VLR in GSM.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

20. Anchor MSC is the MSC visited by the mobile when a call first begins; anchor MSC thus remains unchanged during the call. Throughout the call's duration and regardless of the number of inter-MSC transfers performed by the mobile, the call is routed from the home MSC to the anchor MSC, and then from the anchor MSC to the visited MSC where the mobile is currently located.

21. a) Local recovery
    b) TCP sender awareness of wireless links
    c) Split-connection approaches

Chapter 6 Problems

Problem 1.

Output corresponding to bit d_1 = [-1,1,-1,1,-1,1,-1,1]
Output corresponding to bit d_0 = [1,-1,1,-1,1,-1,1,-1]

Problem 2.

Sender 2 output = [1,-1,1,1,1,-1,1,1]; [1,-1,1,1,1,-1,1,1]

Problem 3.

$$d_2^1 = \frac{1\times1+(-1)\times(-1)+1\times1+1\times1+1\times1+(-1)\times(-1)+1\times1+1\times1}{8} = 1$$

$$d_2^2 = \frac{1\times1+(-1)\times(-1)+1\times1+1\times1+1\times1+(-1)\times(-1)+1\times1+1\times1}{8} = 1$$

Problem 4

Sender 1: (1, 1, 1, -1, 1, -1, -1, -1)
Sender 2: (1, -1, 1, 1, 1, 1, 1, 1)

Problem 5

a) The two APs will typically have different SSIDs and MAC addresses. A wireless station arriving to the café will associate with one of the SSIDs (that is, one of the APs). After association, there is a virtual link between the new station and the AP. Label the APs AP1 and AP2. Suppose the new station associates with AP1. When the new station sends a frame, it will be addressed to AP1. Although AP2 will also receive the frame, it will not process the frame because the frame is not addressed to it. Thus, the two ISPs can work in parallel over the same channel. However, the two ISPs will be sharing the same wireless bandwidth. If wireless stations in different ISPs transmit at the same time, there will be a collision. For 802.11b, the maximum aggregate transmission rate for the two ISPs is 11 Mbps.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

b) Now if two wireless stations in different ISPs (and hence different channels) transmit at the same time, there will not be a collision. Thus, the maximum aggregate transmission rate for the two ISPs is 22 Mbps for 802.11b.

Problem 6

Suppose that wireless station H1 has 1000 long frames to transmit. (H1 may be an AP that is forwarding an MP3 to some other wireless station.) Suppose initially H1 is the only station that wants to transmit, but that while half-way through transmitting its first frame, H2 wants to transmit a frame. For simplicity, also suppose every station can hear every other station's signal (that is, no hidden terminals). Before transmitting, H2 will sense that the channel is busy, and therefore choose a random backoff value.

Now suppose that after sending its first frame, H1 returns to step 1; that is, it waits a short period of times (DIFS) and then starts to transmit the second frame. H1's second frame will then be transmitted while H2 is stuck in backoff, waiting for an idle channel. Thus, H1 should get to transmit all of its 1000 frames before H2 has a chance to access the channel. On the other hand, if H1 goes to step 2 after transmitting a frame, then it too chooses a random backoff value, thereby giving a fair chance to H2. Thus, fairness was the rationale behind this design choice.

Problem 7

A frame without data is 32 bytes long. Assuming a transmission rate of 11 Mbps, the time to transmit a control frame (such as an RTS frame, a CTS frame, or an ACK frame) is (256 bits)/(11 Mbps) = 23 usec. The time required to transmit the data frame is (8256 bits)/(11 Mbps) = 751

DIFS + RTS + SIFS + CTS + SIFS + FRAME + SIFS + ACK
= DIFS + 3SIFS + (3*23 + 751) usec = DIFS + 3SIFS + 820 usec

Problem 8

a) 1 message/ 2 slots
b) 2 messages/slot
c) 1 message/slot
d) i) 1 message/slot
   ii) 2 messages/slot
   iii) 2 messages/slot
e) i) 1 message/4 slots

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

ii) slot 1: Message A→ B, message D→ C
slot 2: Ack B→ A
slot 3: Ack C→ D

= 2 messages/ 3 slots

iii)
slot 1: Message C→ D
slot 2: Ack D→C, message A→ B
slot 3: Ack B→ A

} Repeat

= 2 messages/3 slots

Problem 10

a) 10 Mbps if it only transmits to node A. This solution is not fair since only A is getting served. By "fair" it means that each of the four nodes should be allotted equal number of slots.

b) For the fairness requirement such that each node receives an equal amount of data during each downstream sub-frame, let n_1, n_2, n_3, and n_4 respectively represent the number of slots that A, B, C and D get.
Now,

data transmitted to A in 1 slot = 10t Mbits

(assuming the duration of each slot to be t)

Hence,

Total amount of data transmitted to A (in n_1 slots) = $10t\ n_1$

Similarly total amounts of data transmitted to B, C, and D equal to $5t\ n_2$, $2.5t\ n_3$, and $t\ n_4$ respectively.

Now, to fulfill the given fairness requirement, we have the following condition:

$10t\ n_1 = 5t\ n_2 = 2.5t\ n_3 = t\ n_4$

Hence,

$n_2 = 2\ n_1$
$n_3 = 4\ n_1$
$n_4 = 10\ n_1$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Now, the total number of slots is N. Hence,

$n_1 + n_2 + n_3 + n_4 = N$

i.e. $n_1 + 2\,n_1 + 4\,n_1 + 10\,n_1 = N$

i.e. $n_1 = N/17$

Hence,

$n_2 = 2N/17$

$n_3 = 4N/17$

$n_4 = 10N/17$

The average transmission rate is given by:

$(10t\,n_1 + 5t\,n_2 + 2.5t\,n_3 + t\,n_4)/tN$

$= (10N/17 + 5 * 2N/17 + 2.5 * 4N/17 + 1 * 10N/17)/N$

$= 40/17 = 2.35$ Mbps

c) Let node A receives twice as much data as nodes B, C, and D during the sub-frame.

Hence,

$10tn_1 = 2 * 5tn_2 = 2 * 2.5tn_3 = 2 * tn_4$

i.e. $n_2 = n_1$

$n_3 = 2n_1$

$n_4 = 5n_1$

Again,

$n_1 + n_2 + n_3 + n_4 = N$

i.e. $n_1 + n_1 + 2n_1 + 5n_1 = N$

i.e. $n_1 = N/9$

Now, average transmission rate is given by:

$(10t\,n_1 + 5t\,n_2 + 2.5t\,n_3 + t\,n_4)/tN$

$= 25/9 = 2.78$ Mbps

Similarly, considering nodes B, C, or D receive twice as much data as any other nodes, different values for the average transmission rate can be calculated.

Problem 11

a) No. All the routers might not be able to route the datagram immediately. This is because the Distance Vector algorithm (as well as the inter-AS routing protocols like BGP) is decentralized and takes some time to terminate. So, during the time when the algorithm is still running as a result of advertisements from the new foreign network, some of the routers may not be able to route datagrams destined to the mobile node.

b) Yes. This might happen when one of the nodes has just left a foreign network and joined a new foreign network. In this situation, the routing entries from the old

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

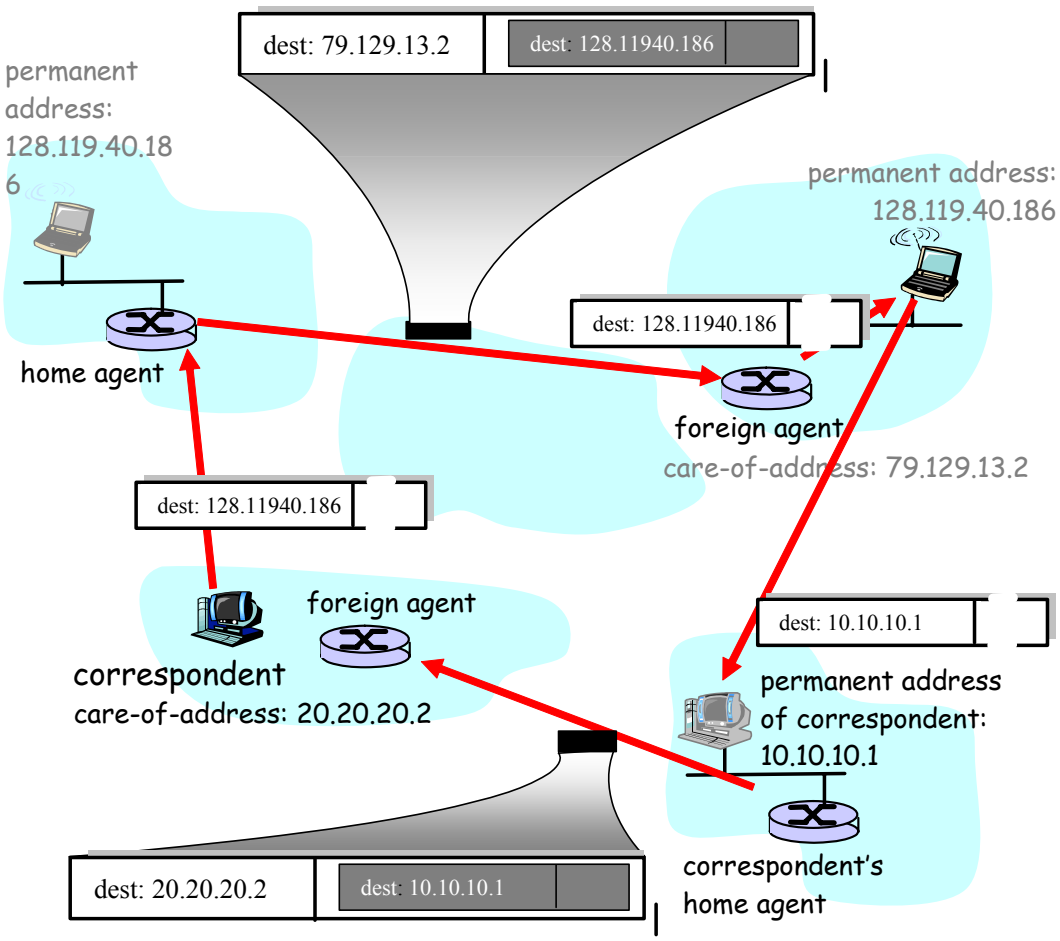
foreign network might not have been completely withdrawn when the entries from the new network are being propagated.

c) The time it takes for a router to learn a path to the mobile node depends on the number of hops between the router and the edge router of the foreign network for the node.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 12



If the correspondent is mobile, then any datagrams destined to the correspondent would have to pass through the **correspondent's home agent**. The **foreign agent** in the network being visited would also need to be involved, since it is this foreign agent that notifies the correspondent's home agent of the location of the correspondent. Datagrams received by the correspondent's home agent would need to be encapsulated/tunneled between the correspondent's home agent and foreign agent, (as in the case of the encapsulated diagram at the top of Figure 6.23.

Problem 13

Because datagrams must be first forward to the home agent, and from there to the mobile, the delays will generally be longer than via direct routing. Note that it *is* possible, however, that the direct delay from the correspondent to the mobile (i.e., if the datagram is not routed through the home agent) could actually be smaller than the sum of the delay

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

from the correspondent to the home agent and from there to the mobile. It would depend on the delays on these various path segments. Note that indirect routing also adds a home agent processing (e.g., encapsulation) delay.

Problem 14

First, we note that chaining was discussed at the end of section 6.5. In the case of chaining using indirect routing through a home agent, the following events would happen:

- The mobile node arrives at A, A notifies the home agent that the mobile is now visiting A and that datagrams to the mobile should now be forwarded to the specified care-of-address (COA) in A.
- The mobile node moves to B. The foreign agent at B must notify the foreign agent at A that the mobile is no longer resident in A but in fact is resident in B and has the specified COA in B. From then on, the foreign agent in A will forward datagrams it receives that are addressed to the mobile's COA in A to the mobile's COA in B.
- The mobile node moves to C. The foreign agent at C must notify the foreign agent at B that the mobile is no longer resident in B but in fact is resident in C and has the specified COA in C. From then on, the foreign agent in B will forward datagrams it receives (from the foreign agent in A) that are addressed to the mobile's COA in B to the mobile's COA in C.

Note that when the mobile goes offline (i.e., has no address) or returns to its home network, the datagram-forwarding state maintained by the foreign agents in A, B and C must be removed. This teardown must also be done through signaling messages. Note that the home agent is not aware of the mobile's mobility beyond A, and that the correspondent is not at all aware of the mobile's mobility.

In the case that chaining is not used, the following events would happen:

- The mobile node arrives at A, A notifies the home agent that the mobile is now visiting A and that datagrams to the mobile should now be forwarded to the specified care-of-address (COA) in A.
- The mobile node moves to B. The foreign agent at B must notify the foreign agent at A and the home agent that the mobile is no longer resident in A but in fact is resident in B and has the specified COA in B. The foreign agent in A can remove its state about the mobile, since it is no longer in A. From then on, the home agent will forward datagrams it receives that are addressed to the mobile's COA in B.
- The mobile node moves to C. The foreign agent at C must notify the foreign agent at B and the home agent that the mobile is no longer resident in B but in fact is resident in C and has the specified COA in C. The foreign agent in B can remove its state about the mobile, since it is no longer in B. From then on, the home agent will forward datagrams it receives that are addressed to the mobile's COA in C.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

When the mobile goes offline or returns to its home network, the datagram-forwarding state maintained by the foreign agent in C must be removed. This teardown must also be done through signaling messages. Note that the home agent is always aware of the mobile's current foreign network. However, the correspondent is still blissfully unaware of the mobile's mobility.

Problem 15

Two mobiles could certainly have the same care-of-address in the same visited network. Indeed, if the care-of-address is the address of the foreign agent, then this address would be the same. Once the foreign agent decapsulates the tunneled datagram and determines the address of the mobile, then separate addresses would need to be used to send the datagrams separately to their different destinations (mobiles) within the visited network.

Problem 16

If the MSRN is provided to the HLR, then the value of the MSRN must be updated in the HLR whenever the MSRN changes (e.g., when there is a handoff that requires the MSRN to change). The advantage of having the MSRN in the HLR is that the value can be provided quickly, without querying the VLR. By providing the address of the VLR Rather than the MSRN), there is no need to be refreshing the MSRN in the HLR.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 7 Review Questions

1. Streaming stored audio/video: pause/resume, re-positioning, fast-forward; real-time interactive audio and video: people communicating and responding in real time.

2. Camp 1: No fundamental changes in TCP/IP protocols; add bandwidth where needed; also use caching, content distribution networks, and multicast overlay networks. Camp 2: Provide a network service that allows applications to reserve bandwidth in the network. Camp 3, differentiated service: introduce simple classifying and policing schemes at the edge of the network, and give different datagrams different levels of service according to their class in the router queues.

3. Uncompressed audio stored on CD has a bit rate of 1411.2 Kbps. mp3 files are typically encoded in 128 Kbps or less, thereby giving them a compression ratio of almost 11. Image compression ratios are in the range of 10 to 100.

4. Figure 6.1: simple, doesn't require meta file or streaming server; Figure 6.2: allows media player to interact directly with the web server, doesn't require a streaming server; Figure 6.3: media player interacts directly with a streaming server, which has been designed for the specific streaming application.

5. End-to-end delay is the time it takes a packet to travel across the network from source to destination. Delay jitter is the fluctuation of end-to-end delay from packet to the next packet.

6. A packet that arrives after its scheduled play out time can not be played out. Therefore, from the perspective of the application, the packet has been lost.

7. First scheme: send a redundant encoded chunk after every n chunks; the redundant chunk is obtained by exclusive OR-ing the n original chunks. Second scheme: send a lower-resolution low-bit rate scheme along with the original stream. Interleaving does not increase the bandwidth requirements of a stream.

8. The role of the DNS is to forward HTTP requests to DNS server managed by the CDN, which in turn redirects the request to an appropriate CDN server. The DNS does not have to be modified to support a CDN. A CDN should provide DNS with the host name and IP address of its authoritative name server (See Section 2.5.3).
9.
   a) Models of traffic demand between network end points
   b) Well-defined performance requirements
   c) Models to predict end-end performance for a given workload model, and techniques to find a minimal high cost bandwidth allocation that will result in all user requirements being met.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

10. RTP streams in different sessions: different multicast addresses; RTP streams in the same session: SSRC field; RTP packets are distinguished from RTCP packets by using distinct port numbers.

11. Reception report packets: includes info about fraction of packets lost, last sequence number, inter-arrival jitter; sender report packets: timestamp and wall clock time of most recently generated RTP packet, number of packets sent, number of bytes sent ; source description packets: e-mail address of the sender, the sender's name, the application that generates the RTP stream.

12. The role of a SIP registrar is to keep track of the users and their corresponding IP addresses which they are currently using. Each SIP registrar keeps track of the users that belong to its domain. It also forwards INVITE messages (for users in its domain) to the IP address which the user is currently using. In this regard, its role is similar to that of an authoritative name server in DNS.

13. In non-preemptive priority queuing, the transmission of a packet is not interrupted once it has begun. In preemptive priority queuing, the transmission of a packet will be interrupted if a higher priority packet arrives before transmission completes. This would mean that portions of the packet would be sent into the network as separate chunks; these chunks would no longer all have the appropriate header fields. For this reason, preemptive priority queuing is not used.

14. A scheduling discipline that is not work conserving is time division multiplexing, whereby a rotating frame is partitioned into slots, with each slot exclusively available to a particular class.

15. FIFO: line at Starbucks. RR: merging traffic (taking 1 vehicle from first lane then 1 from the second, then 1 from the first, and so on). WFQ: ticket counter at airport providing service to 2 people from first class and 1 from economy, and again 2 from first class, and so on.

16. Scalability: Per-flow resource reservation implies the need for a router to process resource reservations and maintain per-flow state for each flow passing through the router. Flexibly service: The Intserv framework provides for a small number of pre-specified service classes.

Chapter 7 Problems

Problem 2

x(t) will ramp up exponentially fast to the TCP available bandwidth due to TCP slow start. Then x(t) will remain roughly constant at the TCP available bandwidth until the client buffer fills. At that time, x(t) will drop to approximately d until the media is sent and the client buffer remains roughly full.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 3

No, they are not the same thing. The client application reads data from the TCP receive buffer and puts it in the client buffer. If the client buffer becomes full, then application will stop reading from the TCP receive buffer until some room opens up in the client buffer.

Problem 4

a) $160 + h$ bytes are sent every 20 msec. Thus the transmission rate is

$$\frac{(160+h)\cdot 8}{20} Kbps = (64 + .4h) Kbps$$

b)

IP header:	20bytes
UDP header:	8bytes
RTP header:	12bytes

h=40 bytes (a 25% increase in the transmission rate!)

Problem 5

a) Denote $d^{(n)}$ for the estimate after the *n*th sample.

$$d^{(1)} = r_4 - t_4$$

$$d^{(2)} = u(r_3 - t_3) + (1-u)(r_4 - t_4)$$

$$d^{(3)} = u(r_2 - t_2) + (1-u)\left[u(r_3 - t_3) + (1-u)(r_4 - t_4)\right]$$

$$= u(r_2 - t_2) + (1-u)u(r_3 - t_3) + (1-u)^2(r_4 - t_4)$$

$$d^{(4)} = u(r_1 - t_1) + (1-u)d^{(3)}$$

$$= u(r_1 - t_1) + (1-u)u(r_2 - t_2) + (1-u)^2 u(r_3 - t_3) + (1-u)^3(r_4 - t_4)$$

b)

$$d^{(n)} = u\sum_{j=1}^{n-1}(1-u)^j(r_j - t_j) + (1-u)^n(r_n - t_n)$$

c)

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

$$d^{(\infty)} = \frac{u}{1-u}\sum_{j=1}^{\infty}(1-u)^j(r_j - t_j)$$

$$= \frac{1}{9}\sum_{j=1}^{\infty}.9^j(r_j - t_j)$$

The weight given to past samples decays exponentially.

Problem 6

a) Denote $v^{(n)}$ for the estimate after the *n*th sample. Let $\Delta_j = r_j - t_j$.

$$v^{(1)} = \left|\ \Delta_4 - d^{(1)}\ \right| \quad (=0)$$

$$v^{(2)} = u\left|\ \Delta_3 - d^{(2)}\ \right| + (1-u)\left|\ \Delta_4 - d^{(1)}\ \right|$$

$$v^{(3)} = u\left|\ \Delta_2 - d^{(3)}\ \right| + (1-u)v^{(2)}$$

$$= u\left|\ \Delta_2 - d^{(3)}\ \right| + u(1-u)\left|\ \Delta_3 - d^{(2)}\ \right| + (1-u)^2\left|\ \Delta_4 - d^{(1)}\ \right|$$

$$v^{(4)} = u\left|\ \Delta_1 - d^{(4)}\ \right| + (1-u)v^{(3)}$$

$$= u\left|\ \Delta_1 - d^{(4)}\ \right| + (1-u)u\left|\ \Delta_2 - d^{(3)}\ \right| + u(1-u)^2\left|\ \Delta_3 - d^{(2)}\ \right|$$

$$+ (1-u)^3\left|\ \Delta_4 - d^{(1)}\ \right|$$

$$= u\left[\ \left|\ \Delta_1 - d^{(4)}\ \right| + (1-u)\left|\ \Delta_2 - d^{(3)}\ \right| + (1-u)^2\left|\ \Delta_3 - d^{(2)}\ \right|\ \right]$$

$$+ (1-u)^3\left|\ \Delta_4 - d^{(1)}\ \right|$$

b)

$$v^{(n)} = u\sum_{j=1}^{n-1}(1-u)^{j-1}\left|\ \Delta_j - d^{(n-j+1)}\ \right| + (1-u)^n\left|\ \Delta_n - d^{(1)}\ \right|$$

Problem 7

a) $r_1 - t_1 + r_2 - t_2 + \ldots + r_{n-1} - t_{n-1} = (n-1)d_{n-1}$

Substituting this into the expression for d_n gives

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

$$d_n = \frac{n-1}{n} d_{n-1} + \frac{r_n - t_n}{n}$$

b) The delay estimate in part (a) is an average of the delays. It gives equal weight to recent delays and to "old" delays. The delay estimate in Section 6.3 gives more weight to recent delays; delays in the distant past have relatively little impact on the estimate.

Problem 8

The two procedures are very similar. They both use the same formula, thereby resulting in exponentially decreasing weights for past samples.

One difference is that for estimating average RTT, the time when the data is sent and when the acknowledgement is received is recorded on the same machine. For the delay estimate, the two values are recorded on different machines. Thus the sample delay can actually be negative.

Problem 9

a) If there is packet lost, then other packets may have been transmitted in between the two received packets.
b) Let S_i denote the sequence number of the *i*th received packet. If

$$t_i - t_{i-1} > 20m\sec$$

and

$$S_i = S_{i-1} + 1$$

then packet *I* begins a new talkspurt.

Problem 10

a) The delay of packet 2 is 7 slots. The delay of packet 3 is 9 slots. The delay of packet 4 is 8 slots. The delay of packet 5 is 7 slots. The delay of packet 6 is 9 slots. The delay of packet 7 is 8 slots. The delay of packet 8 is > 8 slots.

b) Packets 3, 4, 6, 7, and 8 will not be received in time for their playout if playout begins at t-8.

c) Packets 3 and 6 will not be received in time for their playout if playout begins at t=9.

d) No packets will arrive after their playout time if playout time begins at t=10.

Problem 11

The answers to parts a and b are in the table below:

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Packet Number	$r_i - t_i$	d_i	v_i
1	7	7	0
2	8	7.10	0.09
3	8	7.19	0.162
4	7	7.17	0.163
5	9	7.35	0.311
6	9	7.52	0.428
7	8	7.57	0.429
8	8	7.61	0.425

Problem 12

a) Both schemes require 25% more bandwidth. The first scheme has a playback delay of 5 packets. The second scheme has a delay of 2 packets.

b) The first scheme will be able to reconstruct the original high-quality audio encoding. The second scheme will use the low quality audio encoding for the lost packets and will therefore have lower overall quality.

c) For the first scheme, many of the original packets will be lost and audio quality will be very poor. For the second scheme, every audio chunk will be available at the receiver, although only the low quality version will be available for every other chunk. Audio quality will be acceptable.

Problem 13

The CDN company provides a mechanism so that when a client requests content, the content is provided by the CDN server that can best deliver the content to the specific client. The server may be the closest CDN server to the client (perhaps in the same ISP as the client) or may be a CDN server with a congestion-free path to the client. In this way, even if a CDN does not increase the amount of link capacity in the network, the performance seen by the hosts is improved.

Problem 14

Yes. If the path between the requesting host and the CDN server, chosen by the CDN, is already too congested or if the chosen CDN server is already too busy, then it might be possible that the host experiences worse performance that what it would have if it contacted the remote server directly.

Problem 15

The inter-arrival jitter *J* is defined to be the mean deviation (smoothed absolute value) of the difference *D* in packet spacing at the receiver compared to the sender for a pair of packets. As shown in the equation below, this is equivalent to the "relative transit time"

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

for the two packets; the relative transit time is the difference between a packet's RTP timestamp and the receiver's clock at the time of arrival. If T_i is the RTP timestamp for packet i and r_i is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, D is defined as

$$D(i, j) = (r_i - r_j) - (s_i - s_j) = (r_j - s_j) - (r_i - s_i)$$

The inter-arrival jitter is calculated continuously as each data packet i is received, using this difference D for that packet and the previous packet i-1 in order of arrival (not necessarily in sequence), according to the formula

$$J = J + \frac{|D(i, j) - J|}{16}$$

Whenever a reception report is issued, the current value of J is sampled.

Problem16

a) As discussed in Chapter 2, UDP sockets are identified by the two-tuple consisting of destination IP address and destination port number. So the two packets will indeed pass through the same socket.

b) Yes, Alice only needs one socket. Bob and Claire will choose different SSRC's, so Alice will be able distinguish between the two streams. Another question we could have asked is: How does Alice's software know which stream (i.e. SSRC) belongs to Bob and which stream belongs to Alice? Indeed, Alice's software may want to display the sender's name when the sender is talking. Alice's software gets the SSRC to name mapping from the RTCP source description reports.

Problem 17

a) The session bandwidth is 4* 100 kbps = 400 kbps. Five percent of the session bandwidth is 20 kbps.

b) Sends each user is both a sender and receiver, each user gets 5 kbps for RTCP packets (receiver reports, sender reports, and source description packets).

Problem 18

a) Like HTTP, all request and response methods are in ASCII text. RTSP also has methods (SETUP, PLAY, PAUSE), and the server responds with standardized reply codes. Yes, using the GET method, HTTP can be used to request a stream.

b) RTSP messages use different port numbers than the media streams. Thus RTSP is out-of-band. HTTP objects are sent within the HTTP response message. Thus HTTP is in-band. HTTP does not keep session state: each request is handled independently. RTSP uses the Session ID to maintain session state. For example, in the lab (programming assignment) for this chapter, the RTSP server is in one several states for each session ID. When in the pause state, the server stores the number of the frame at which the pause occurred.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 19

a) True
b) True
c) No, RTP streams can be sent to/from any port number. See the SIP example in Section 6.4.3
d) No, typically they are assigned different SSRC values.
e) True
f) False, she is indicating that she wishes to *receive* GSM audio
g) False, she is indicating that she wishes to *receive* audio on port 48753
h) True, 5060 for both source and destination port numbers
i) True
j) False, this is a requirement of H.323 and not SIP.

Problem 20

a) One possible sequence is 1 2 1 3 1 2 1 3 1 2 1 3 …
Another possible sequence is 1 1 2 1 1 3 1 1 2 1 1 3 1 1 2 1 1 3 …
b) 1 1 3 1 1 3 1 1 3 1 1 3 …

Problem 21

a)

Packet	Time leaving the queue	Delay
1	0	0
2	1	1
3	2	1
4	3	2
5	5	2
6	4	2
7	6	3
8	7	2
9	8	3
10	9	2
11	10	2
12	11	3
Average Delay		**1.91**

b)

Packet	Time leaving the queue	Delay
1	0	0
2	2	2
3	1	0

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

4	6	5
5	4	1
6	7	5
7	3	0
8	9	4
9	5	0
10	10	3
11	8	0
12	11	3
Average Delay		**1.91**

c)

Packet	Time leaving the queue	Delay
1	0	0
2	2	2
3	4	3
4	1	0
5	3	0
6	6	4
7	5	2
8	7	2
9	9	4
10	11	4
11	8	0
12	10	2
Average Delay		**1.91**

d)

Packet	Time leaving the queue	Delay	Note
1	0	0	WFQ
2	2	2	WFQ
3	1	0	WFQ
4	5	4	WFQ
5	3	0	WFQ
6	7	5	Idealized WFQ scheduling
7	4	1	WFQ
8	9	4	WFQ
9	6	1	Idealized WFQ scheduling
10	10	3	WFQ
11	8	0	WFQ
12	11	3	WFQ
Average Delay		**1.91**	

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

e) It can be noticed that the average delay for all four cases is the same (1.91 seconds).

Problem 22

a)

Packet	Time leaving the queue	Delay
1	0	0
2	4	4
3	5	4
4	1	0
5	3	0
6	2	0
7	6	3
8	9	4
9	7	2
10	10	3
11	8	0
12	11	3
Average Delay		**1.91**

b)

Packet	Time leaving the queue	Delay
1	0	0
2	1	1
3	3	2
4	2	1
5	6	3
6	4	2
7	5	2
8	9	4
9	7	2
10	10	3
11	8	0
12	11	3
Average Delay		**1.91**

c)

Packet	Time leaving the queue	Delay
1	0	0
2	1	1
3	2	1
4	3	2

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

5	9	6
6	6	4
7	4	1
8	7	2
9	5	0
10	8	1
11	11	3
12	10	2
Average Delay		**1.91**

Problem 23

Time Slot	Packets in the queue	Number of tokens in bucket
0	1, 2, 3	2
1	3, 4	1
2	4,5	1
3	5,6	1
4	6	1
5	-	1
6	7, 8	2
7	9, 10	1
8	10	1

Time Slot	Packets in output buffer
0	1, 2
1	3
2	4
3	5
4	6
5	-
6	7, 8
7	9
8	10

Problem 24

Time Slot	Packets in the queue	Number of tokens in bucket
0	1, 2, 3	2
1	3, 4	2
2	5	2
3	6	2
4	-	2
5	-	2
6	7, 8	2
7	9, 10	2

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

8	-	2

Time Slot	Packets in output buffer
0	1, 2
1	3, 4
2	5
3	6
4	-
5	-
6	7, 8
7	9, 10
8	-

Problem 25

No. The answer still remains the same as in Problem 24.

Problem 26

See figure below. For the second leaky bucket, $r = p, b = 1$.

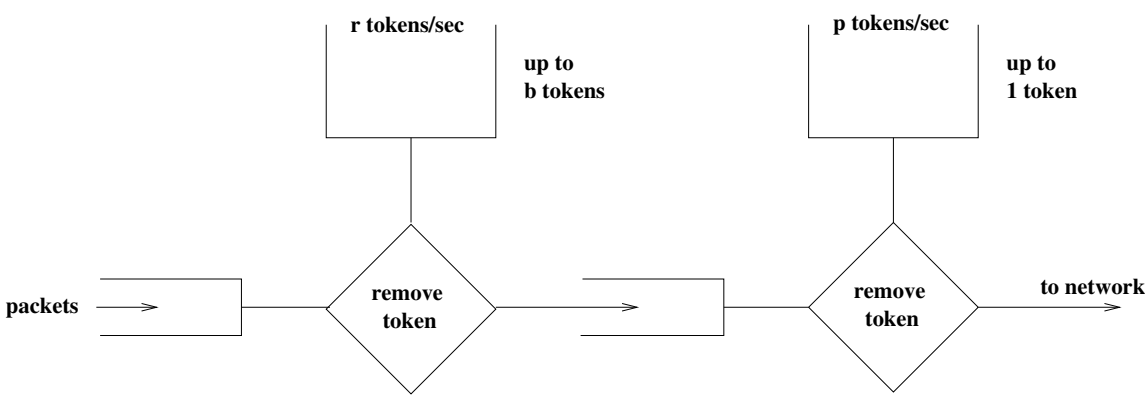


Figure: Solution to problem 26

Problem 27

No.

Problem 28

Let τ be a time at which flow 1 traffic starts to accumulate in the queue. We refer to τ as the beginning of a flow-1 busy period. Let $t > \tau$ be another time in the same flow-1 busy period. Let $T_1(\tau, t)$ be the amount of flow-1 traffic transmitted in the interval $[\tau, t]$. Clearly,

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

$$T_1(\tau, t) \geq \frac{W_1}{\sum W_j} R(t-\tau)$$

Let $Q_1(t)$ be the amount of flow-1 traffic in the queue at time *t*. Clearly,

$$Q_1(t) = b_1 + r_1(t-\tau) - T_1(\tau, t)$$

$$\leq b_1 + r_1(t-\tau) + \frac{W_1}{\sum W_j} R(t-\tau)$$

$$= b_1 + (t-\tau)\left[r_1 - \frac{W_1}{\sum W_j} R\right]$$

Since $r_1 < \frac{W_1}{\sum W_j} R$, $Q_1(t) \leq b_1$. Thus the maximum amount of flow-1 traffic in the queue is b_1. The minimal rate at which this traffic is served is $\frac{W_1 R}{\sum W_j}$.

Thus, the maximum delay for a flow-1 bit is

$$\frac{b_1}{W_1 R / \sum W_j} = d_{\max}.$$

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Chapter 8 Review Questions

Question 1.
Confidentiality is the property that the original plaintext message can not be determined by an attacker who intercepts the ciphertext-encryption of the original plaintext message. Message integrity is the property that the receiver can detect whether the message sent (whether encrypted or not) was altered in transit. The two are thus different concepts, and one can have one without the other. An encrypted message that is altered in transmit may still be confidential (the attacker can not determine the original plaintext) but will not have message integrity if the error is undetected. Similarly, a message that is altered in transit (and detected) could have been sent in plaintext and thus would not be confidential.

Question 2.
A passive intruder only monitors ("sniffs", intercepts) messages. An active intruder can also monitor traffic, but will also actively send messages into the network

Question 3.
One important difference between symmetric and public key systems is that in symmetric key systems both the sender and receiver must know the same (secret) key. In public key systems, the encryption and decryption keys are distinct. The encryption key is known by the entire world (including the sender), but the decryption key is known only by the receiver.

Question 4.
In this case, a known plaintext attack is performed. If, somehow, the message encrypted by the sender was chosen by the attacker, then this would be a chosen-plaintext attack.

Question 5.
If each user wants to communicate with N other users, then each pair of users must have a shared symmetric key. There are N*(N-1)/2 such pairs and thus there are N*(N-1)/2 keys. With a public key system, each user has a public key which is known to all, and a private key (which is secret and only known by the user). There are thus 2N keys in the public key system.

Question 6.
A nonce is used to ensure that the person being authenticated is "live." Nonces thus are used to combat playback attacks.

Question 7.
Once in a lifetimes means that the entity sending the nonce will never again use that value to check whether another entity is "live".

Question 8.
In a man-in-the-middle attacker, the attacker interposes him/herself between the sender and receiver, often performing some transformation (e.g., re-encoding or altering) of data

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

between the sender and receiver. Man-in-the-middle attacks can be particularly pernicious since the sender and receiver will each receive what the other has sent and since they are using encryption would think that they have achieved confidentiality.

Question 9.
Suppose Bob sends an encrypted document to Alice. To be verifiable, Alice must be able to convince herself that Bob sent the encrypted document. To be non-forgeable, Alice must be able to convince herself that only Bob could have sent the encrypted document (e.g.,, non one else could have guess a key and encrypted/sent the document) To be non-reputiable, Alice must be able to convince someone else that only Bob could have sent the document. To illustrate the latter distinction, suppose Bob and Alice share a secret key, and they are the only ones in the world who know the key. If Alice receives a document that was encrypted with the key, and knows that she did not encrypt the document herself, then the document is known to be verifiable and non-forgeable (assuming a suitably strong encryption system was used). However, Alice can *not* convince someone else that Bob must have sent the document, since in fact Alice knew the key herself and could have encrypted/sent the document.

Question 10.
One requirement of a message digest is that given a message M, it is very difficult to find another message M' that has the same message digest and, as a corollary, that given a message digest value it is difficult to find a message M" that has that given message digest value. We have "message integrity" in the sense that we have reasonable confidence that given a message M and its signed message digest that the message was not altered since the message digest was computed and signed. This is not true of the Internet checksum, where we saw in Figure 7.18 that it easy to find two messages with the same Internet checksum.

Question 11.
A public-key signed message digest is "better" in that one need only encrypt (using the private key) a short message digest, rather than the entire message. Since public key encryption with a technique like RSA is expensive, it's desirable to have to sign (encrypt) a smaller amount of data than a larger amount of data.

Question 12

No. This is because a hash function is a one-way function. That is, given any hash value, the original message cannot be recovered (given h such that h=H(m), one cannot recover m from h).

Question 13

A certification authority binds an individual's identity with a public key. The CA signs that key with its (the CAs) private key. Thus, given the public key of a CA, one can retrieve the CA-signed public key for an entity, verify the CA's signature, and then have the CA-certified public key for an entity.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Question 14.
The AH provides for authentication and message integrity, while ESP provides for authentication, integrity, and confidentiality.

Chapter 8 Problems

Problem 1.
The encoding of "This is an easy problem" is "uasi si my cmiw lokngch".
The decoding of "rmij'u uamu xyj" is "wasn't that fun".

Problem 2.
If Trudy knew that the words "bob" and "alice" appeared in the text, then she would know the ciphertext for b,o,a,l,i,c,e (since "bob" is the only palindrome in the message, and "alice" is the only 5-letter word. If Trudy knows the ciphertext for 7 of the letters, then she only needs to try 19!, rather than 26!, plaintext-ciphertext pairs. The difference between 19! and 26! is 26*25*24...*20, which is 3315312000, or approximately 10^9.

Problem 3.
Every letter in the alphabet appears in the phrase "The quick fox jumps over the lazy brown dog." Given this phrase in a chosen plaintext attack (where the attacker has both the plain text, and the ciphertext), the Caesar cipher would be broken - the intruder would know the ciphertext character for every plaintext character. However, the Vigenere cipher does not alway translate a given plaintext character to the same ciphertext character each time, and hence a Vigenere cipher would not be immediately broken by this chosen plaintext attack.

Problem 4.
We are given $p = 3$ and $q = 11$. We thus have $n = 33$ and $q = 11$. Choose $e = 9$ (it might be a good idea to give students a hint that 9 is a good value to choose, since the resulting calculations are less likely to run into numerical stability problems than other choices for *e.*) since 3 and $(p-1)*(q-1) = 20$ have no common factors. Choose $d = 9$ also so that $e*d = 81$ and thus $e*d-1 = 80$ is exactly divisible by 20. We can now perform the RSA encryption and decryption using $n = 33$, $e = 9$ and $d = 9$.

```
letter  m      m**e          ciphertext = m**e mod 33
  h     8      134217728     29
  e     5      1953125       20
  l     12     5159780352    12
  l     12     5159780352    12
  o     15     38443359375   3

ciphertext    c**d             m = c**d mod n   letter
29           14507145975869    8                h
20           512000000000      5                e
12           5159780352        12               l
```

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

12	5159780352	12	l
3	19683	15	o

Problem 5.

Bob does not know if he is talking to Trudy or Alice initially. Bob and Alice share a secret key K_{A-B} that is unknown to Trudy. Trudy wants Bob to authenticate her (Trudy) as Alice. Trudy is going to have Bob authenticate himself, and waits for Bob to start:

1. **Bob-to-Trudy: "I am Bob"** Commentary: Bob starts to authenticate himself. Bob's authentication of himself to the other side then stops for a few steps.
2. **Trudy-to-Bob: "I am Alice"** Commentary: Trudy starts to authenticate herself as Alice
3. **Bob-to-Trudy: "R"** Commentary: Bob responds to step 2 by sending a nonce in reply. Trudy does not yet know $K_{A-B}(R)$ so she can not yet reply.
4. **Trudy-to-Bob: "R"** Commentary: Trudy responds to step 1 now continuing Bob's authentication, picking as the nonce for Bob to encrypt, *the exact same value that Bob sent her to encrypt in Step 3.*
5. **Bob-to-Trudy: "**$K_{A-B}(R)$" Bob completes his own authentication of himself to the other side by encrypting the nonce he was sent in step 4. Trudy now has $K_{A-B}(R)$. (Note: she does not have, nor need, K_{A-B}
6. *Trudy-to-Bob:* **"**$K_{A-B}(R)$" Trudy completes her authentication, responding to the R that Bob sent in step 3 above with $K_{A-B}(R)$. Since Trudy has returned the properly encrypted nonce that Bob send in step 3, Bob thinks Trudy is Alice!

Problem 6.

This wouldn't really solve the problem. Just as Bob thinks (incorrectly) that he is authenticating Alice in the first half of Figure 7.14, so too can Trudy fool Alice into thinking (incorrectly) that she is authenticating Bob. The root of the problem that neither Bob nor Alice can tell is the public key they are getting is indeed the public key of Alice of Bob.

Problem 7.

A full blown encryption is more computationally complex than a message digest such as MD5.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

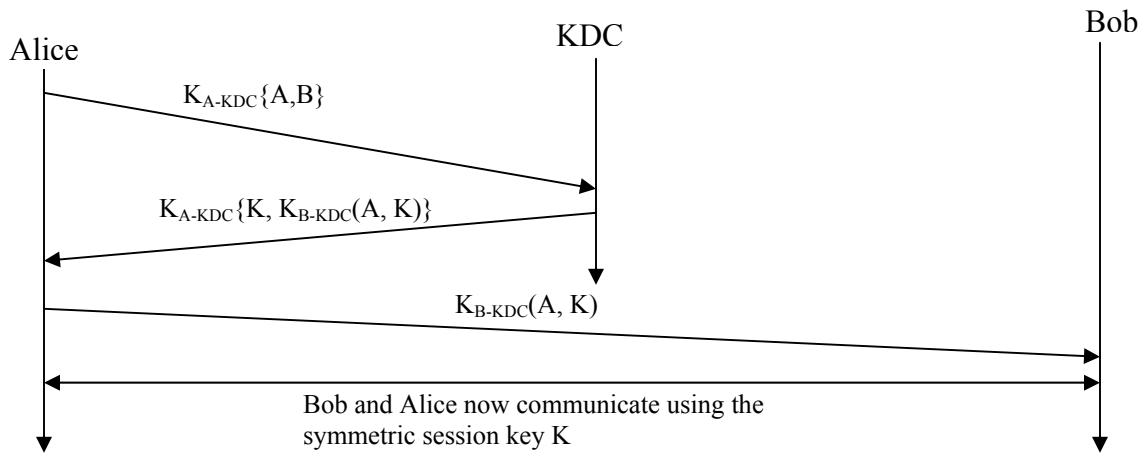
Problem 8

The message

I	O	U	1
9	0	.	9
0	B	O	B

has the same checksum

Problem 9



Problem 10

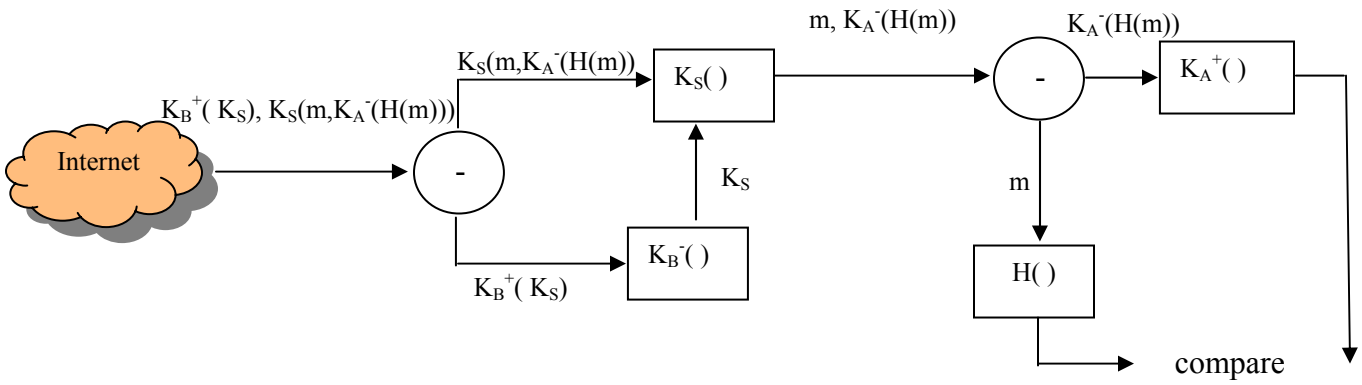


Figure: Operations performed by Bob for confidentiality, integrity, and authentication

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Problem 11

a) False
b) True
c) False
d) False
e) True
f) False

Problem 12

a) Since IV = 11, the key stream is 111110100000 …….

Given, m = 10100000
Hence, ICV = 1010 XOR 0000 = 1010
The three fields will be:

IV: 11
Encrypted message: 10100000 XOR 11111010 = 01011010
Encrypted ICV: 1010 XOR 0000 = 1010

b) Receiver extracts the IV (11) and generates the key stream 111110100000 …….

XORs the encrypted message with key stream to recover the original message:
01011010 XOR 11111010 = 10100000

XORs the encrypted ICV with key stream to recover the original ICV:

1010 XOR 0000 = 1010

Now, XORs the first 4 bits of recovered message with its last 4 bits:
1010 XOR 0000 = 1010 (which equals the recovered ICV)

c) Since the ICV is calculated as the XOR of first 4 bits of message with last 4 bits of message, either the 1st bit or the 5th bit of the message has to be flipped for the received packet to pass the ICV check.

d) For part (a), the encrypted message was 01011010
Flipping the 1st bit gives, 11011010
Trudy XORs this message with the key stream:
11011010 XOR 11111010 = 00100000

If Trudy flipped the first bit of the encrypted ICV, the ICV value received by the receiver is 0010
The receiver XORs this value with the key stream to get the ICV:
0010 XOR 0000 = 0010

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

The receiver now calculates the ICV from the recovered message:
0010 XOR 0000 = 0010 (which equals the recovered ICV and so the received packet passes the TCV check)

Problem 13

Filter Table:

Action	Source address	Dest address	Protocol	Source port	Dest port	Flag bit	Check connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	22	any	
allow	outside of 222.22/16	222.22/16	TCP	22	> 1023	ACK	x
Allow	outside of 222.22/16	222.22.0.12	TCP	>1023	80	any	
Allow	222.22.0.12	outside of 222.22/16	TCP	80	>1023	any	
deny	All	all	all	all	all	all	

Connection Table:

Source address	Dest address	Source port	Dest port
222.22.1.7	37.96.87.123	12699	22
222.22.93.2	199.1.205.23	37654	22
222.22.65.143	203.77.240.43	48712	22

Problem 14

a) Client
b) IP: 216.75.194.220, port: 443
c) 283
d) 3 SSL records
e) Yes, it contains an encrypted master secret
f) First byte: bc; Last byte: 29
g) 6

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

h) 9

Chapter 9 Review Questions

Question 1.
A network manager would like to have network management capabilities when (a) a component of the network fails, (b) a component of the network is about to fail, and is acting "flaky" (c) a component of the network has been compromised from a security standpoint and is attacking the network, e.g., by launching a DOS attack by flooding the network with packets, (d) traffic levels exceed a certain threshold on a link, causing packets to be dropped, (e) everything is running smoothly (in order to *know* that everything is running smoothly and there are no problems). There are many additional reasons as well.

Question 2.
Performance management, fault management, configuration management, accounting management, security management.

Question 3.
Network management is more narrowly defined, as it focuses on the resources in the network – monitoring their functions and controlling their operation. These resources are combined (used) in various ways to implement services. Note that while the network resources may all be functioning as they should, they may not be sufficient to implement a service with a given level of performance; this latter concern is an aspect of service management.

Question 4.
- ***Managing entity:*** control the collection, processing, analysis, display of network management information, and is used by the network manager to control the devices in the network.
- ***Managed device:*** a piece of network equipment that is under the control of the managing entity.
- ***Management agent:*** a software process running on a managed device that communicated with the managing entity and takes action on the managed device under the control of the managing entity.
- ***MIB:*** pieces of information associated with all of the managed objects in a device.
- ***Network management protocol:*** runs between the managing entity of the management agents on the managed devices, allowing the agents to alert the managing entity to potential problems, and allowing the managing entity to send commands to the management agents.

Question 5.
The SMI is a data-definition language used to defined the pieces of information in an SNMP MIB.

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Question 6.
The trap message is sent by the management agent to the managing entity (and requires no response from the managing entity). A request-response message is sent by the managing entity, with the response coming back from the management agent.

Question 7.
GetRequest, GetNextRequest, GetBulkRequest, SetRequest, InformRequest, Response, Trap

Question 8.
The SNMP engine is the part of an SNMP implementation that handles the dispatching, processing, authentication, access control, and timeliness of he SNMP messages. See Figure 8.5.

Question 9.
The ASN.1 object identifier tree provides a standard way to name objects.

Question 10.
The role of the presentation layer is to allow the sending and receiving of data in a machine-independent format (i.e., without regard to the particular storage and architectural conventions of the sender and receiver).

Question 11.
The Internet does not have presentation layer. It is up to the implementers of protocols to make sure that data is sent and received in the desired format.

Question 12.
In TLV encoding, each piece of data is tagged with its type, length, and value.

Chapter 9 Problems

Problem 1.
Request response mode will generally have more overhead (measured in terms of the number of messages exchanged) for several reasons. First, each piece of information received by the manager requires two messages: the poll and the response. Trapping generates only a single message to the sender. If the manager really only wants to be notified when a condition occurs, polling has more overhead, since many of the polling messages may indicate that the waited-for condition has not yet occurred. Trapping generates a message only when the condition occurs.

Trapping will also immediately notify the manager when an event occurs. With polling, the manager needs will need to wait for half a polling cycle (on average) between when the event occurs and the manager discovers (via its poll message) that the event has occurred.

Downloaded by Amit Sharma (amitmudgal69@gmail.com)

If a trap message is lost, the managed device will not send another copy. If a poll message, or its response, is lost the manager would know there has been a lost message (since the reply never arrives). Hence the manager could repoll, if needed.

Problem 2.
Often, the time when network management is most needed is in times of stress, when the network may be severely congested and packets are being lost. With SNMP running over TCP, TCP's congestion control would cause SNMP to back-off and stop sending messages at precisely the time when the network manager needs to send SNMP messages.

Problem 3.
1.3.6.1.2.1.5

Problem 4.
Microsoft file formats are under 1.2.840.113556.4 (see section 8.3.2)

Problem 6.

45Julia22 ‘00000001’ ‘00010100’

Problem 7

45Child 21 ‘10100000’

This document is available free of charge on studocu

Downloaded by Amit Sharma (amitmudgal69@gmail.com)