*Computer Networking: A Top-Down Approach,*

*7th Edition*

Solutions to Review Questions and Problems

Version Date: July 2016

This document contains the solutions to review questions and problems for the 7th edition of *Computer Networking: A Top-Down Approach* 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.

*Acknowledgments:* Over the years, several students and colleagues have helped us prepare this solutions manual. Special thanks goes to Honggang Zhang, Rakesh Kumar, Prithula Dhungel, and Vijay Annapureddy. Also thanks to all the readers who have made suggestions and corrected errors.

All material © copyright 1996-2016 by J.F. Kurose and K.W. Ross. All rights reserved

# 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, PDAs, Internet-connected game consoles, etc.
2. From Wikipedia: Diplomatic protocol is commonly described as a set of international courtesy rules. These well-established and time-honored rules have made it easier for nations and people to live and work together. Part of protocol has always been the acknowledgment of the hierarchical standing of all present. Protocol rules are based on the principles of civility.
3. Standards are important for protocols so that people can create networking systems and products that interoperate.
4. 1. Dial-up modem over telephone line: home; 2. DSL over telephone line: home or small office; 3. Cable to HFC: home; 4. 100 Mbps switched Ethernet: enterprise; 5. Wifi (802.11): home and enterprise: 6. 3G and 4G: wide-area wireless.
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. In most American cities, the 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.

8. Today, Ethernet most commonly runs over twisted-pair copper wire. It also can run over fibers optic links.

9. Dial up modems: up to 56 Kbps, bandwidth is dedicated; ADSL: up to 24 Mbps downstream and 2.5 Mbps upstream, bandwidth is dedicated; HFC, rates up to 42.8 Mbps and upstream rates of up to 30.7 Mbps, bandwidth is shared. FTTH: 2-10Mbps upload; 10-20 Mbps download; bandwidth is not shared.

10. There are two popular wireless Internet access technologies today:

1. Wifi (802.11) In a wireless LAN, wireless users transmit/receive packets to/from an base station (i.e., 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.
2. 3G and 4G wide-area wireless access networks. 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. At time t0 the sending host begins to transmit. At time *t1 = L/R1*, 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 *t1*. At time *t2 = t1 + L/R2*, 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/R1 + L/R2*.

12. 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. FDM requires sophisticated analog hardware to shift signal into appropriate frequency bands.

13. 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

d) Probability that all three users are transmitting simultaneously = 

=(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.

14. If the two ISPs do not peer with each other, then when they send traffic to each other they have to send the traffic through a provider ISP (intermediary), to which they have to pay for carrying the traffic. By peering with each other directly, the two ISPs can reduce their payments to their provider ISPs. An Internet Exchange Points (IXP) (typically in a standalone building with its own switches) is a meeting point where multiple ISPs can connect and/or peer together. An ISP earns its money by charging each of the the ISPs that connect to the IXP a relatively small fee, which may depend on the amount of traffic sent to or received from the IXP.

15. Google's private network connects together all its data centers, big and small. Traffic between the Google data centers passes over its private network rather than over the public Internet. Many of these data centers are located in, or close to, lower tier ISPs. Therefore, when Google delivers content to a user, it often can bypass higher tier ISPs. What motivates content providers to create these networks? First, the content provider has more control over the user experience, since it has to use few intermediary ISPs. Second, it can save money by sending less traffic into provider networks. Third, if ISPs decide to charge more money to highly profitable content providers (in countries where net neutrality doesn't apply), the content providers can avoid these extra payments.

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. a) 1000 km, 1 Mbps, 100 bytes

b) 100 km, 1 Mbps, 100 bytes

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

19. a) 500 kbps

b) 64 seconds

c) 100kbps; 320 seconds

20. End system A breaks the large file into chunks. It adds header to each chunk, thereby generating multiple packets from the file. The header in each packet includes the IP address of the destination (end system B). The packet switch uses the destination IP address in the packet 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 destination address.

21. The maximum emission rate is 500 packets/sec and the maximum transmission rate is

350 packets/sec. The corresponding traffic intensity is 500/350 =1.43 > 1. Loss will eventually occur for each experiment; but the time when loss first occurs will be different from one experiment to the next due to the randomness in the emission process.

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 network, link and physical layers (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 Transport layer as well.) Link layer switches process link and physical layers (layers 1 through2). Hosts process all five layers.

26. a) Virus

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.

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

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

### At time N\*(L/R) the first packet has reached the destination, the second packet is stored in the last router, the third packet is stored in the next-to-last router, etc. At time N\*(L/R) + L/R, the second packet has reached the destination, the third packet is stored in the last router, etc. Continuing with this logic, we see that at time N\*(L/R) + (P-1)\*(L/R) = (N+P-1)\*(L/R) all packets have reached the destination.

### Problem 3

a) A circuit-switched network would be well suited to the application, 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 without significant waste. In addition, the overhead costs of setting up and tearing down connections are amortized over the lengthy duration of a typical application session.

b) In the worst case, all the applications simultaneously transmit over one or more network links. However, since each link has sufficient bandwidth to handle the sum of all of the applications' data rates, no congestion (very little queuing) will occur. Given such generous link capacities, the network does not need congestion control mechanisms.

### Problem 4

1. Between the switch in the upper left and the switch in the upper right we can have 4 connections. Similarly we can have four connections between each of the 3 other pairs of adjacent switches. Thus, this network can support up to 16 connections.
2. We can *4* connections passing through the switch in the upper-right-hand corner and another *4* connections passing through the switch in the lower-left-hand corner, giving a total of *8* connections.
3. Yes. For the connections between A and C, we route two connections through B and two connections through D. For the connections between B and D, we route two connections through A and two connections through C. In this manner, there are at most 4 connections passing through any link.

### Problem 5

Tollbooths are 75 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 2 minutes, for the first tollbooth to service the 10 cars. Each of these cars has a propagation delay of 45 minutes (travel 75 km) before arriving at the second tollbooth. Thus, all the cars are lined up before the second tollbooth after 47 minutes. The whole process repeats itself for traveling between the second and third tollbooths. It also takes 2 minutes for the third tollbooth to service the 10 cars. Thus the total delay is 96 minutes.

b) Delay between tollbooths is 8\*12 seconds plus 45 minutes, i.e., 46 minutes and 36 seconds. The total delay is twice this amount plus 8\*12 seconds, i.e., 94 minutes and 48 seconds.

### Problem 6

a)  seconds.

b)  seconds.

c)  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

km.

### Problem 7

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

sec=7msec.

The time required to transmit the packet is

sec=sec.

Propagation delay = 10 msec.

The delay until decoding is

7msec +sec + 10msec = 17.224msec

A similar analysis shows that all bits experience a delay of 17.224 msec.

### Problem 8

a) 20 users can be supported.

b) .

c) .

d) .

We use the central limit theorem to approximate this probability. Let  be independent random variables such that .

“21 or more users”







when  is a standard normal r.v. Thus “21 or more users”.

### Problem 9

1. 10,000
2. 

### Problem 10

The first end system requires *L/R1* to transmit the packet onto the first link; the packet propagates over the first link in *d1/s1*; the packet switch adds a processing delay of *dproc*; after receiving the entire packet, the packet switch connecting the first and the second link requires *L/R2* to transmit the packet onto the second link; the packet propagates over the second link in *d2/s2*. Similarly, we can find the delay caused by the second switch and the third link: *L/R3*, *dproc*, and *d3/s3*.

Adding these five delays gives

*dend-end = L/R1 + L/R2 + L/R3 + d1/s1 + d2/s2 + d3/s3+ dproc+ dproc*

To answer the second question, we simply plug the values into the equation to get 6 + 6 + 6 + 20+16 + 4 + 3 + 3 = 64 msec.

### Problem 11

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

*dend-end = L/R + d1/s1 + d2/s2+ d3/s3*

For the values in Problem 10, we get 6 + 20 + 16 + 4 = 46 msec.

### Problem 12

The arriving packet must first wait for the link to transmit 4.5 \*1,500 bytes = 6,750 bytes or 54,000 bits. Since these bits are transmitted at 2 Mbps, the queuing delay is 27 msec. Generally, the queuing delay is (*nL* + (*L* - *x*))/*R*.

### Problem 13

1. 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 + ....... + (N-1)L/R)/N*

*= L/(RN) \* (1 + 2 + ..... + (N-1))*

*= L/(RN) \* N(N-1)/2*

*= LN(N-1)/(2RN)*

*= (N-1)L/(2R)*

Note that here we used the well-known fact:

*1 + 2 + ....... + N = N(N+1)/2*

1. It takes  seconds to transmit the  packets. Thus, the buffer is empty when a each batch of  packets arrive. Thus, the average delay of a packet across all batches is the average delay within one batch, i.e., (*N-*1)*L*/*2R*.

### Problem 14

1. The transmission delay is . The total delay is



1. Let .

Total delay = 

For x=0, the total delay =0; as we increase x, total delay increases, approaching infinity as x approaches 1/a.

### Problem 15

Total delay .

### Problem 16

The total number of packets in the system includes those in the buffer and the packet that is being transmitted. So, N=10+1.

Because , so (10+1)=a\*(queuing delay + transmission delay). That is,

11=a\*(0.01+1/100)=a\*(0.01+0.01). Thus, a=550 packets/sec.

### Problem 17

1. There are  nodes (the source host and the  routers). Let denote the processing delay at the th node. Let  be the transmission rate of the th link and let

. Let  be the propagation delay across the th link. Then

.

1. Let  denote the average queuing delay at node . Then

.

### Problem 18

On linux you can use the command

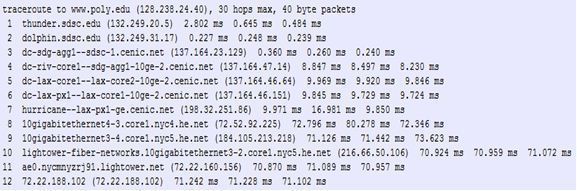
traceroute www.targethost.com

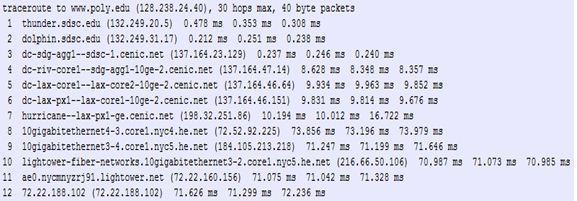
and in the Windows command prompt you can use

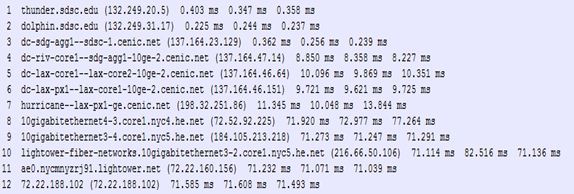
tracert www.targethost.com

In either case, you will get three delay measurements. For those three measurements you can calculate the mean and standard deviation. Repeat the experiment at different times of the day and comment on any changes.

Here is an example solution:

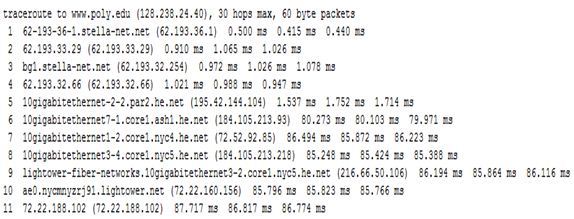


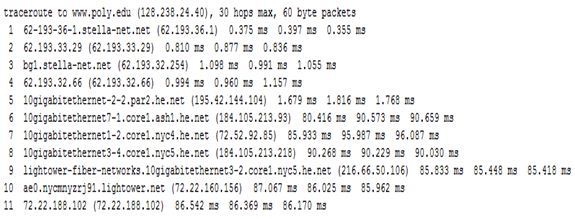


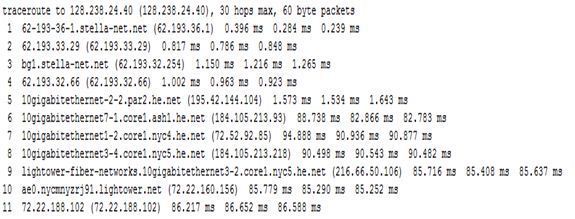


Traceroutes between San Diego Super Computer Center and [www.poly.edu](http://www.poly.edu)

1. The average (mean) of the round-trip delays at each of the three hours is 71.18 ms, 71.38 ms and 71.55 ms, respectively. The standard deviations are 0.075 ms, 0.21 ms, 0.05 ms, respectively.
2. In this example, the traceroutes have 12 routers in the path at each of the three hours. No, the paths didn’t change during any of the hours.
3. Traceroute packets passed through four ISP networks from source to destination. Yes, in this experiment the largest delays occurred at peering interfaces between adjacent ISPs.





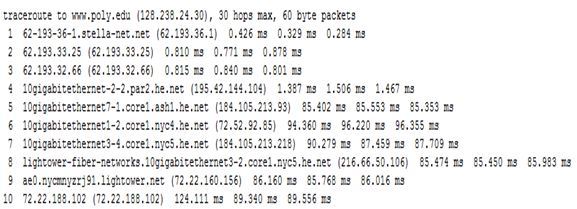


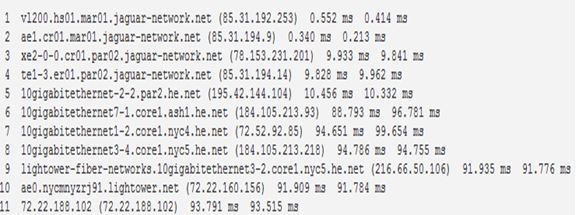
Traceroutes from [www.stella-net.net](http://www.stella-net.net) (France) to [www.poly.edu](http://www.poly.edu) (USA).

1. The average round-trip delays at each of the three hours are 87.09 ms, 86.35 ms and 86.48 ms, respectively. The standard deviations are 0.53 ms, 0.18 ms, 0.23 ms, respectively. In this example, there are 11 routers in the path at each of the three hours. No, the paths didn’t change during any of the hours. Traceroute packets passed three ISP networks from source to destination. Yes, in this experiment the largest delays occurred at peering interfaces between adjacent ISPs.

### Problem 19

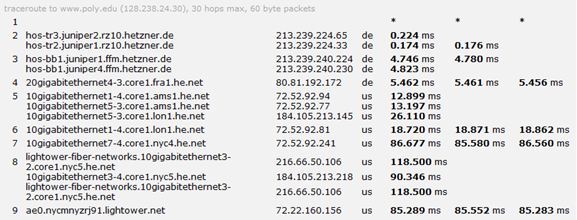
An example solution:

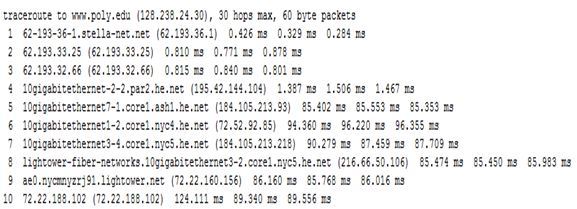




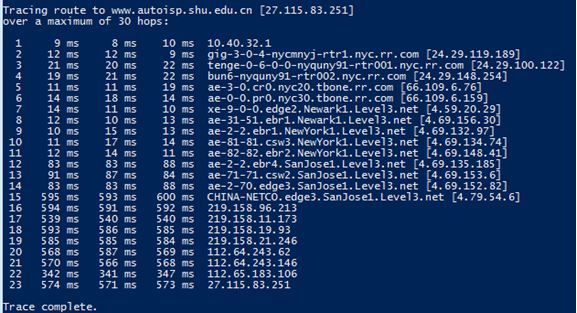
Traceroutes from two different cities in France to New York City in United States

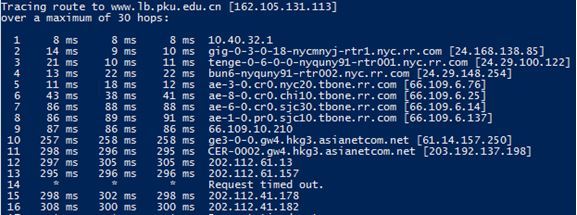
1. In these traceroutes from two different cities in France to the same destination host in United States, seven links are in common including the transatlantic link.





1. In this example of traceroutes from one city in France and from another city in Germany to the same host in United States, three links are in common including the transatlantic link.





Traceroutes to two different cities in China from same host in United States

1. Five links are common in the two traceroutes. The two traceroutes diverge before reaching China

### Problem 20

Throughput = *min{Rs, Rc, R/M}*

### Problem 21

If only use one path, the max throughput is given by:

.

If use all paths, the max throughput is given by .

### Problem 22

Probability of successfully receiving a packet is: ps= (1-p)N.

The number of transmissions needed to be performed until the packet is successfully received by the client is a geometric random variable with success probability ps. Thus, the average number of transmissions needed is given by: 1/ps . Then, the average number of re-transmissions needed is given by: 1/ps -1.

### Problem 23

Let’s call the first packet A and call the second packet B.

1. If the bottleneck link is the first link, then packet B is queued at the first link waiting for the transmission of packet A. So the packet inter-arrival time at the destination is simply *L/Rs*.
2. If the second link is the bottleneck link and both packets are sent back to back, it must be true that the second packet arrives at the input queue of the second link before the second link finishes the transmission of the first packet. That is,

*L/Rs + L/Rs + dprop < L/Rs + dprop + L/Rc*

The left hand side of the above inequality represents the time needed by the second packet to *arrive at* the input queue of the second link (the second link has not started transmitting the second packet yet). The right hand side represents the time needed by the first packet to finish its transmission onto the second link.

If we send the second packet *T* seconds later, we will ensure that there is no queuing delay for the second packet at the second link if we have:

*L/Rs + L/Rs + dprop + T >= L/Rs + dprop + L/Rc*

Thus, the minimum value of T is *L/Rc* − *L/Rs* .

### Problem 24

40 terabytes = 40 \* 1012 \* 8 bits. So, if using the dedicated link, it will take 40 \* 1012 \* 8 / (100 \*106 ) =3200000 seconds = 37 days. But with FedEx overnight delivery, you can guarantee the data arrives in one day, and it should cost less than $100.

### Problem 25

1. 160,000 bits
2. 160,000 bits
3. The bandwidth-delay product of a link is the maximum number of bits that can be in the link.
4. the width of a bit = length of link / bandwidth-delay product, so 1 bit is 125 meters long, which is longer than a football field
5. s/R

### Problem 26

*s*/*R*=20000km, then *R*=*s*/20000km= 2.5\*108/(2\*107)= 12.5 bps

### Problem 27

1. 80,000,000 bits
2. 800,000 bits, this is because that the maximum number of bits that will be in the link at any given time = min(bandwidth delay product, packet size) = 800,000 bits.
3. .25 meters

### Problem 28

1. *ttrans + tprop* = 400 msec + 80 msec = 480 msec.
2. 20 \* (*ttrans + 2 tprop*) = 20\*(20 msec + 80 msec) = 2 sec.
3. Breaking up a file takes longer to transmit because each data packet and its corresponding acknowledgement packet add their own propagation delays.

### Problem 29

Recall geostationary satellite is 36,000 kilometers away from earth surface.

1. 150 msec
2. 1,500,000 bits
3. 600,000,000 bits

### Problem 30

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

### Problem 31

1. Time to send message from source host to first packet switch = With store-and-forward switching, the total time to move message from source host to destination host = 
2. Time to send 1st packet from source host to first packet switch = . . Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = 
3. Time at which 1st packet is received at the destination host = . After this, every 5msec one packet will be received; thus time at which last (800th) packet is received = . It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).
4. Without message segmentation, if bit errors are not tolerated, if there is a single bit error, the whole message has to be retransmitted (rather than a single packet).
5. Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.
6. Packets have to be put in sequence at the destination.
7. 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 32

Yes, the delays in the applet correspond to the delays in the Problem 31.The propagation delays affect the overall end-to-end delays both for packet switching and message switching equally.

### 

### Problem 33

There are *F*/*S* packets. Each packet is S=80 bits. Time at which the last packet is received at the first router is sec. At this time, the first F/S-2 packets are at the destination, and the F/S-1 packet is at the second router. The last packet must then be transmitted by the first router and the second router, with each transmission taking sec. Thus delay in sending the whole file is

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



**Problem 34**

The circuit-switched telephone networks and the Internet are connected together at "gateways". When a Skype user (connected to the Internet) calls an ordinary telephone, a circuit is established between a gateway and the telephone user over the circuit switched network. The skype user's voice is sent in packets over the Internet to the gateway. At the gateway, the voice signal is reconstructed and then sent over the circuit. In the other direction, the voice signal is sent over the circuit switched network to the gateway. The gateway packetizes the voice signal and sends the voice packets to the Skype user.

# Chapter 2 Review Questions

1. The Web: HTTP; file transfer: FTP; remote login: Telnet; e-mail: SMTP; BitTorrent file sharing: BitTorrent protocol
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. 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 socket in the destination process.
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. One such example is remote word processing, for example, with Google docs. However, because Google docs runs over the Internet (using TCP), timing guarantees are not provided.
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) Confidentiality (via encryption)

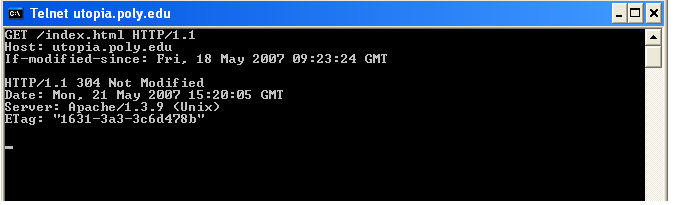
Neither

1. 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.
2. 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.
3. 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.
4. When the user first visits the site, the server creates a unique identification number, creates an entry in its back-end database, and returns this identification number as 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.

1. Web caching can bring the desired content “closer” to the user, possibly 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.
2. Telnet is not available in Windows 7 by default. to make it available, go to Control Panel, Programs and Features, Turn Windows Features On or Off, Check Telnet client. To start Telnet, in Windows command prompt, issue the following command

> telnet webserverver 80

where "webserver" is some webserver. After issuing the command, you have established a TCP connection between your client telnet program and the web server. Then type in an HTTP GET message. An example is given below:



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

1. A list of several popular messaging apps: WhatsApp, Facebook Messenger, WeChat, and Snapchat. These apps use the different protocols than SMS.
2. The message is first 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](mailto: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](mailto: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 messages 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-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”.

1. 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).
2. 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.
3. You should be able to see the sender's IP address for a user with an .edu email address. But you will not be able to see the sender's IP address if the user uses a gmail account.
4. 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 provides chunks to Bob throughout a 30-second interval.
5. Recall that in BitTorrent, a peer picks a random peer and optimistically unchokes the peer for a short period of time. Therefore, Alice will eventually be optimistically unchoked by one of her neighbors, during which time she will receive chunks from that neighbor.
6. 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.
7. One server placement philosophy is called Enter Deep, which enter deep into the access networks of Internet Service Providers, by deploying server clusters in access ISPs all over the world. The goal is to reduce delays and increase throughput between end users and the CDN servers. Another philosophy is Bring Home, which bring the ISPs home by building large CDN server clusters at a smaller number of sites and typically placing these server clusters in IXPs (Internet Exchange Points). This Bring Home design typically results in lower maintenance and management cost, compared with the enter-deep design philosophy.
8. Other than network-related factors, there are some important factors to consider, such as load-balancing (clients should not be directed to overload clusters), diurnal effects, variations across DNS servers within a network, limited availability of rarely accessed video, and the need to alleviate hot-spots that may arise due to popular video content.

Reference paper:

Torres, Ruben, et al. "Dissecting video server selection strategies in the YouTube CDN." The 31st IEEE International Conference on. Distributed Computing Systems (ICDCS), 2011.

Another factor to consider is ISP delivery cost – the clusters may be chosen so that specific ISPs are used to carry CDN-to-client traffic, taking into account the different cost structures in the contractual relationships between ISPs and cluster operators.

1. 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.
2. 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

e) F

### Problem 2

SMS (Short Message Service) is a technology that allows the sending and receiving of text messages between mobile phones over cellular networks. One SMS message can contain data of 140 bytes and it supports languages internationally. The maximum size of a message can be 160 7-bit characters, 140 8-bit characters, or 70 16-bit characters. SMS is realized through the Mobile Application Part (MAP) of the SS#7 protocol, and the Short Message protocol is defined by 3GPP TS 23.040 and 3GPP TS 23.041. In addition, MMS (Multimedia Messaging Service) extends the capability of original text messages, and support sending photos, longer text messages, and other content.

iMessage is an instant messenger service developed by Apple. iMessage supports texts, photos, audios or videos that we send to iOS devices and Macs over cellular data network or WiFi. Apple’s iMessage is based on a proprietary, binary protocol APNs (Apple Push Notification Service).

WhatsApp Messenger is an instant messenger service that supports many mobile platforms such as iOS, Android, Mobile Phone, and Blackberry. WhatsApp users can send each other unlimited images, texts, audios, or videos over cellular data network or WiFi. WhatsApp uses the XMPP protocol (Extensible Messaging and Presence Protocol).

iMessage and WhatsApp are different than SMS because they use data plan to send messages and they work on TCP/IP networks, but SMS use the text messaging plan we purchase from our wireless carrier. Moreover, iMessage and WhatsApp support sending photos, videos, files, etc., while the original SMS can only send text message. Finally, iMessage and WhatsApp can work via WiFi, but SMS cannot.

### Problem 3

Application layer protocols: DNS and HTTP

Transport layer protocols: UDP for DNS; TCP for HTTP

### Problem 4

1. 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.
2. The browser is running HTTP version 1.1, as indicated just before the first <cr><lf> pair.
3. The browser is requesting a persistent connection, as indicated by the Connection: keep-alive.
4. 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 that carried the HTTP GET request) to answer this question.
5. Mozilla/5.0. The browser type information is needed by the server to send different versions of the same object to different types of browsers.

### Problem 5

1. 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 2008 12:39:45 Greenwich Mean Time.
2. The document index.html was last modified on Saturday 10 Dec 2005 18:27:46 GMT.
3. There are 3874 bytes in the document being returned.
4. 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

1. 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 connection-token "close" in the Connection-header field of the http request/reply.
2. HTTP does not provide any encryption services.
3. (From RFC 2616) “Clients that use persistent connections should limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy.”
4. Yes. (From RFC 2616) “A client might have started to send a new request at the same time that the server has decided to close the "idle" connection. From the server's point of view, the connection is being closed while it was idle, but from the client's point of view, a request is in progress.”

### Problem 7

The total amount of time to get the IP address is

.

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



### Problem 8



.





1. Persistent connection with pipelining. This is the default mode of HTTP.



.

Persistent connection without pipelining, without parallel connections.



.

### Problem 9

1. 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 = (850,000 bits)/(15,000,000 bits/sec) = .0567 sec

The traffic intensity on the link is given by βΔ=(16 requests/sec)(.0567 sec/request) = 0.907. Thus, the average access delay is (.0567 sec)/(1 - .907) ≈ .6 seconds. The total average response time is therefore .6 sec + 3 sec = 3.6 sec.

1. The traffic intensity on the access link is reduced by 60% since the 60% of the requests are satisfied within the institutional network. Thus the average access delay is (.0567 sec)/[1 – (.4)(.907)] = .089 seconds. The response time is approximately zero if the request is satisfied by the cache (which happens with probability .6); the average response time is .089 sec + 3 sec = 3.089 sec for cache misses (which happens 40% of the time). So the average response time is (.6)(0 sec) + (.4)(3.089 sec) = 1.24 seconds. Thus the average response time is reduced from 3.6 sec to 1.24 sec.

### Problem 10

Note that each downloaded object can be completely put into one data packet. Let Tp denote the one-way propagation delay between the client and the server.

First consider parallel downloads using non-persistent connections. Parallel downloads would allow 10 connections to share the 150 bits/sec bandwidth, giving each just 15 bits/sec. Thus, the total time needed to receive all objects is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p )

+ (200/(150/10)+*T*p + 200/(150/10) +*T*p + 200/(150/10)+*T*p + 100,000/(150/10)+ *T*p )

= 7377 + 8\**T*p (seconds)

Now consider a persistent HTTP connection. The total time needed is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p )

+ 10\*(200/150+*T*p + 100,000/150+ *T*p )

=7351 + 24\**T*p (seconds)

Assuming the speed of light is 300\*106 m/sec, then Tp=10/(300\*106)=0.03 microsec. Tp is therefore negligible compared with transmission delay.

Thus, we see that persistent HTTP is not significantly faster (less than 1 percent) than the non-persistent case with parallel download.

### Problem 11

1. Yes, because Bob has more connections, he can get a larger share of the link bandwidth.
2. Yes, Bob still needs to perform parallel downloads; otherwise he will get less bandwidth than the other four users.

### Problem 12

Server.py

from socket import \*

serverPort=12000

serverSocket=socket(AF\_INET,SOCK\_STREAM)

serverSocket.bind(('',serverPort))

serverSocket.listen(1)

connectionSocket, addr = serverSocket.accept()

while 1:

sentence = connectionSocket.recv(1024)

print 'From Server:', sentence, '\n' serverSocket.close()

### Problem 13

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 14

SMTP uses a line containing only a period to mark the end of a message body.

HTTP uses “Content-Length header field” to indicate the length of a message body.

No, HTTP cannot use the method used by SMTP, because HTTP message could be binary data, whereas in SMTP, the message body must be in 7-bit ASCII format.

### Problem 15

MTA stands for Mail Transfer Agent. A host sends the message to an MTA. The message then follows a sequence of MTAs to reach the receiver’s mail reader. We see that this spam message follows a chain of MTAs. An honest MTA should report where it receives the message. Notice that in this message, “asusus-4b96 ([58.88.21.177])” does not report from where it received the email. Since we assume only the originator is dishonest, so “asusus-4b96 ([58.88.21.177])” must be the originator.

### Problem 16

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 user's mailbox. This command is useful for “download and keep”. By maintaining a file that lists the messages retrieved during earlier sessions, the client can use the UIDL command to determine which messages on the server have already been seen.

### Problem 17

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

1. C: list

S: 1 498

S: 2 912

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 18

* 1. For a given input of domain name (such as ccn.com), IP address or network administrator name, the *whois* database can be used to locate the corresponding registrar, whois server, DNS server, and so on.
  2. NS4.YAHOO.COM from www.register.com; NS1.MSFT.NET from ww.register.com
  3. *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)

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.

* 1. 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 19

1. The following delegation chain is used for gaia.cs.umass.edu

a.root-servers.net

E.GTLD-SERVERS.NET

ns1.umass.edu(authoritative)

First command:

dig +norecurse @a.root-servers.net any gaia.cs.umass.edu

;; AUTHORITY SECTION:

edu. 172800 IN NS E.GTLD-SERVERS.NET.

edu. 172800 IN NS A.GTLD-SERVERS.NET.

edu. 172800 IN NS G3.NSTLD.COM.

edu. 172800 IN NS D.GTLD-SERVERS.NET.

edu. 172800 IN NS H3.NSTLD.COM.

edu. 172800 IN NS L3.NSTLD.COM.

edu. 172800 IN NS M3.NSTLD.COM.

edu. 172800 IN NS C.GTLD-SERVERS.NET.

Among all returned edu DNS servers, we send a query to the first one.

dig +norecurse @E.GTLD-SERVERS.NET any gaia.cs.umass.edu

umass.edu. 172800 IN NS ns1.umass.edu.

umass.edu. 172800 IN NS ns2.umass.edu.

umass.edu. 172800 IN NS ns3.umass.edu.

Among all three returned authoritative DNS servers, we send a query to the first one.

dig +norecurse @ns1.umass.edu any gaia.cs.umass.edu

gaia.cs.umass.edu. 21600 IN A 128.119.245.12

1. The answer for google.com could be:

a.root-servers.net

E.GTLD-SERVERS.NET

ns1.google.com(authoritative)

### Problem 20

We can periodically take a snapshot of the DNS caches in the local DNS servers. The Web server that appears most frequently in the DNS caches is the most popular server. This is because if more users are interested in a Web server, then DNS requests for that server are more frequently sent by users. Thus, that Web server will appear in the DNS caches more frequently.

For a complete measurement study, see:

Craig E. Wills, Mikhail Mikhailov, Hao Shang

“Inferring Relative Popularity of Internet Applications by Actively Querying DNS Caches”, in IMC'03, October 27­29, 2003, Miami Beach, Florida, USA

### Problem 21

Yes, we can use dig to query that Web site in the local DNS server.

For example, “dig cnn.com” will return the query time for finding cnn.com. If cnn.com was just accessed a couple of seconds ago, an entry for cnn.com is cached in the local DNS cache, so the query time is 0 msec. Otherwise, the query time is large.

### Problem 22

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

*Dcs = max {NF/us, F/dmin}*

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



Where, *F* = 15 Gbits = 15 \* 1024 Mbits

*us* = 30 Mbps

*dmin* = *di*= 2 Mbps

Note, 300Kbps = 300/1024 Mbps.

Client Server

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | N | | |
| 10 | 100 | 1000 |
| u | 300 Kbps | 7680 | 51200 | 512000 |
| 700 Kbps | 7680 | 51200 | 512000 |
| 2 Mbps | 7680 | 51200 | 512000 |

Peer to Peer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | N | | |
| 10 | 100 | 1000 |
| u | 300 Kbps | 7680 | 25904 | 47559 |
| 700 Kbps | 7680 | 15616 | 21525 |
| 2 Mbps | 7680 | 7680 | 7680 |

### Problem 23

1. Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of a rate of *us*/*N*. Note that this rate is less than each of the client’s download rate, since by assumption *us*/*N* ≤ *d*min. Thus each client can also receive at rate *us*/*N*. Since each client receives at rate *us*/*N*, the time for each client to receive the entire file is F/( *us*/*N*) = *NF/ us.* Since all the clients receive the file in *NF/ us*, the overall distribution time is also *NF/ us.*
2. 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 *us*, since by assumption *us*/*N* ≥ *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*.*
3. From Section 2.6 we know that

*DCS* ≥ max {*NF/us, F/d*min} (Equation 1)

Suppose that *us*/*N* ≤ *d*min. Then from Equation 1 we have *DCS* ≥ *NF/us .* But from (a) we have *DCS* ≤ *NF/us .* Combining these two gives:

*DCS* = *NF/us* when *us*/*N* ≤ *d*min. (Equation 2)

We can similarly show that:

*DCS* =*F/d*min when *us*/*N* ≥ *d*min (Equation 3).

Combining Equation 2 and Equation 3 gives the desired result.

### Problem 24

1. Define u = u1 + u2 + ….. + uN. By assumption

us <= (us + u)/N Equation 1

Divide the file into N parts, with the ith part having size (ui/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 + ….. + *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)ri. 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 ui.

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



Thus each peer receives the file in F/us.

1. Again define *u* = *u*1 + *u*2 + ….. + *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 ri. Each peer i forwards the bits arriving at rate ri to each of the other N-1 peers. Additionally, the server sends bits from the (*N+1*) st part at rate rN+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+ …. + *r*N + *N* *r*N+1 = *u*/(*N-1*) + *u*s – *u*/(*N-1*) = *u*s

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



Thus each peer receives the file in NF/(us+u).

(For simplicity, we neglected to specify the size of the file part for i = 1, …., N+1. We now provide that here. Let Δ = (us+u)/N be the distribution time. For i = 1, …, N, the ith file part is Fi = ri Δ bits. The (N+1)st file part is FN+1 = rN+1 Δ bits. It is straightforward to show that F1+ ….. + FN+1 = F.)

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



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

### Problem 25

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

### Problem 26

Yes. His first claim is possible, as long as there are enough peers staying in the swarm for a long enough time. Bob can always receive data through optimistic unchoking by other peers.

His second claim is also true. He can run a client on each host, let each client “free-ride,” and combine the collected chunks from the different hosts into a single file. He can even write a small scheduling program to make the different hosts ask for different chunks of the file. This is actually a kind of Sybil attack in P2P networks.

### **Problem 27**

### a. N files, under the assumption that we do a one-to-one matching by pairing video versions with audio versions in a decreasing order of quality and rate.

### b. 2N files.

### Problem 28

1. 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.
2. 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.
3. 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 29

In the original program, 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 additional 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. 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

**Problem 30**

Yes, you can configure many browsers to open multiple simultaneous connections to a Web site. The advantage is that you will you potentially download the file faster. The disadvantage is that you may be hogging the bandwidth, thereby significantly slowing down the downloads of other users who are sharing the same physical links.

**Problem 31**

For an application such as remote login (telnet and ssh), a byte-stream oriented protocol is very natural since there is no notion of message boundaries in the application. When a user types a character, we simply drop the character into the TCP connection.

In other applications, we may be sending a series of messages that have inherent boundaries between them. For example, when one SMTP mail server sends another SMTP mail server several email messages back to back. Since TCP does not have a mechanism to indicate the boundaries, the application must add the indications itself, so that receiving side of the application can distinguish one message from the next. If each message were instead put into a distinct UDP segment, the receiving end would be able to distinguish the various messages without any indications added by the sending side of the application.

**Problem 32**

To create a web server, we need to run web server software on a host. Many vendors sell web server software. However, the most popular web server software today is Apache, which is open source and free. Over the years it has been highly optimized by the open-source community.