

>>> SOLUTIONS <<<

Welcome to the Final Exam for *Computer Networks*. Read each problem carefully. There are ten required problems (each problem is worth 10 points). There is also an additional extra credit question worth 10 points. You may have with you a calculator, pencils, erasers, blank paper, and one 8.5 x 11 inch "formula sheet". On this formula sheet you may have anything you want (definitions, formulas, homework answers, old exam answers, etc.) as **handwritten by you** on both sides of the sheet. Photocopies, scans, or computer generated/printed text are not allowed on this sheet. You have 120 minutes for the exam. Please use a separate sheet of paper for each question. Good luck and be sure to show your work!

Problem #1

Answer the following questions on the basics of computer networks and the Internet.

- a) Sketch the five-layer Internet protocol stack that we have been using in the class (and is used in Kurose and Ross)

2pts total - 0.5 pts per layer

Application	5
Transport	4
Network	3
Link	2
Physical	1

Points off are "rounded-up". For example, if in Problem 1(a) one layer is wrong, then it is -1. If two layers are wrong it is still -1. If three layers are wrong it is -2. And, so on.

- b) Define protocol. Define interface.

2pts total - 1 pt each item.

A protocol is a complete set of rules for information interchange between same level layers at different sites. An interface is a complete set of rules for information interchange between adjacent layers in one site.

- c) What are the four factors of packet delay?

2pts total - 0.25 pts per item

Processing delay, transmission delay, propagation delay, and queueing (or buffering) delay

- d) What are the two possible causes of packet loss? In the wired Internet, which is the most common cause of packet loss?

2pts total - 1 pt per item

Electrical noise and buffer overflow. Wired networks generally have little noise, so buffer overflow is the predominant factor for packet loss in the wired Internet.

- e) Name a tool (or utility) that can be used to measure delay in the Internet.

2pts total

Ping (also traceroute) can be used to measure round-trip time for a packet.

Problem #2

Answer the following questions related to the Application layer.

- a) An application layer protocol defines how an application's processes, running in different end systems, pass messages to each other. In particular, an application-layer protocol defines four things. What are these four things?

2pts total - 0.5 pts each item

The four things are 1) types of messages, 2) syntax of messages, 3) semantics of messages, and 4) rules for determining when to send messages (timing).

- b) Sketch the packet format of an HTTP packet on Ethernet. Show all packet headers and trailers. You do not need to show the contents of the headers or trailers.

2pts total - 0.5 pts field

```
+-----+-----+-----+-----+-----+-----+
| Ethernet hdr | IP hdr | TCP hdr | HTTP hdr | appl data | Ethernet trlr |
+-----+-----+-----+-----+-----+-----+
```

- c) Name three email access protocols (that is, from user agent to SMTP server) for email download.

2pts total - 1 pt each item

POP, IMAP, and HTTP

- d) What is DNS? Describe what it does and (at a very high level) how it works.

2pts total - 1 pt for what and 1 pt for how

DNS is the Domain Name System and is the Internet-wide "phone book" that associates a host name with an IP address. DNS is a hierarchy of servers. A query for a hostname to IP address resolution travels up the hierarchy of DNS servers until a match is found. Lower-level DNS servers can cache hostname to IP address mappings.

- e) P2P is an application for sharing files. P2P can have a centralized directory of what files are available and where they are accessible (e.g., Napster) or be completely decentralized (e.g., Gnutella). Briefly describe the trade-offs of centralized versus distributed P2P directories.

2pts total - 1 pt for bottleneck and 1 pt for scalability

A centralized directory is simple, but offers a single point of failure and can be a performance bottleneck. A decentralized approach (as used by Gnutella) requires flooding of query messages, which is unscalable (due to overwhelming query traffic).

Problem #3

The appendix contains `tcpServer.c` with some "bugs". Identify the bugs and explain how to fix them. The program compiles successfully for Microsoft Windows. The bugs are, I think, fairly major in scope.

1 pt each

The following are the bugs:

- No `WSAStartup()` after line 27
- Wrong protocol family in line 29 (should be streams, not datagram)
- Wrong socket is in bind in line 39 (should be `welcome_s`)
- No `listen()` after line 45
- Only 2 bytes being copied in `memcpy` in line 56 (should be 4 bytes)

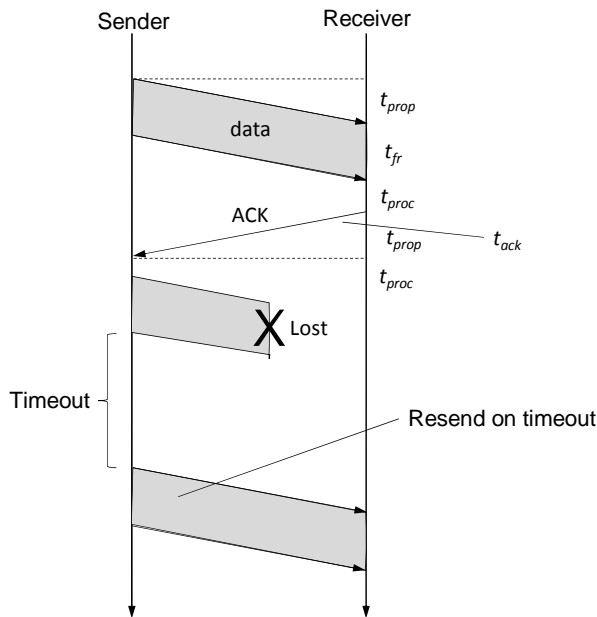
- No ntohs() in printf in line 59
- No "+ 1" for eol character in line 62
- Extra "+ 1" in line 69
- No closesocket() for connect_s after line 82
- No WSACleanup() after line 82

Problem #4

Stop-and-Wait (SAW) is a simple protocol for reliable delivery of packets from source to destination. Answer the following questions regarding SAW.

- a) Sketch the timing diagram for SAW showing both error free and data packet loss cases.

2 pts - roughly 0.5 pts per transmission, delay component, and loss and time-out case



- b) Identify all the delay components of SAW (for example, as shown in your answer to (a)).

2 pts - 0.5 pts each component

The delay components are processing (t_{proc}) twice, transmission of data frame (t_{tr}), transmission of ACK (t_{ack}), and propagation (t_{prop}) twice.

- c) From your answer to (b), derive the formula for Utilization (U) for SAW.

2 pts

$$U = t_{tr} / (t_{tr} + t_{ack} + 2t_{proc} + 2t_{prop})$$

- d) The moon is about 240,000 miles from earth. What would the utilization (U) be of a SAW protocol between the earth and moon that uses 500 byte frames and operates on a point-to-point 1 Mb/s wireless channel? If the channel data rate was increased to 100 Mb/s, what would U be? Assume that packet loss never occurs.

2 pts (anything around 1% is fine assuming formula and calculations are correct)

Remember that speed of light is 1 nanosecond per foot, or close to 5 microseconds per mile. So the t_{prop} to the moon is going to be about 1.2 seconds. The variable t_{tr} is 0.004 seconds. Clearly t_{ack} and t_{proc} are going to be negligible. Thus, $U = 0.004 / (0.004 + 2 * 1.2) = 0.166\%$ (really, really bad!). If the data rate was increased by 100 then t_{tr} would be reduced by 100 giving us $= 0.00004 / (0.00004 + 2 * 1.2) = 0.00166\%$ (really, really, really bad).

- e) Is there a better reliable data transfer protocol than SAW with regards to achieving a higher U ? If so, what is it and what does it do?

2 pts - 1 pt for identifying SW and 1 pt for explaining what it does

In the Sliding Window (SW) protocol multiple data frames (call the window size) can be outstanding (i.e., in flight) before an ACK is required to be received (which when received increments the “window” of frames that can be outstanding).

Problem #5

Answer the following questions about ARP, IPv4, and IPv6

- a) What is the purpose of ARP? Describe how ARP works.

2 pts

The purpose of ARP is to resolve IP addresses to MAC addresses within a subnet. ARP uses MAC-level broadcast where only the host with the matching IP address responds to the host sending the ARP. From this response, the sending host can learn the MAC address of the responding host.

- b) IP addresses are split into netid and hosted portions. Why?

2 pts

This is to simplify (reduce the size of) router tables; router tables contain only netids. Thus, routers route only to the destination network. Within the network (and using the ARP cache) is the final forwarding of the packet to the destination host.

- c) List and describe the fields in an IPv4 packet header.

4pts total - 1/3 pts each (and round up)

An IPv4 header is 20 bytes, the fields are:

- Ver = version
- IHL = Internet header length
- Type = type of service (not used)
- Total length = total length of data unit length
- Identifier = unique value to identify this datagram
- Flags = indicates if fragmentation allowed
- Fragment offset = where fragment belongs
- TTL = time to live
- Protocol = type of next protocol
- Header checksum – to be able to detect errors in the header
- Source and destination address – IP addresses that are 32 bits each
- Options = sender optional stuff

- d) Why was IPv6 created? Describe the most significant changes of IPv6 compared to IPv4.

2 pts

IPv6 was primarily created to overcome the address shortage of IPv4 addresses. The most significant change is the increase in address length from 32 bits to 128 bits. Another change was the addition of a flow label.

Problem #6

Answer the following questions about LANs and Ethernet.

- a) What is a LAN?

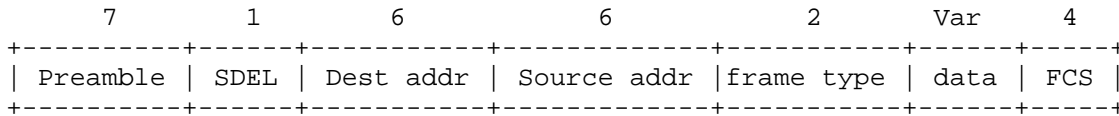
2 pts total

A LAN is a data network optimized for a medium sized area (10s to 100s of stations and 100s to 1000s of meters span). A LAN is owned and operated by a single organization.

b) What are the fields of an IEEE 802.3 Ethernet frame (state how many bytes in each field)?

3 pts total - 0.5 pts each field and for description of frame size

Number of bytes:



An Ethernet frame must be between 64 and 1500 (1518 to be precise) bytes in length. The data field is padded if the needed to achieve the minimum frame size of 64 bytes.

c) Precisely describe (i.e., give the algorithms) CSMA/CD and BEB as used in IEEE 802.3 Ethernet

5 pts total - roughly 1 pt for each line

Ethernet uses a 1-persistent CSMA algorithm:

- (1) if medium is idle then transmit
- (2) if medium is busy then wait until idle and then transmit
- (3) if detect a collision during my transmission immediately halt transmission, transmit a brief jam signal, and wait a random period of time, then (1)

The random period of time is determined using the BEB algorithm:

```

while (attempts < backoff_limit)
k := min(attempts, 10)
r := rand(0, 2^k)
delay := r * slot_time

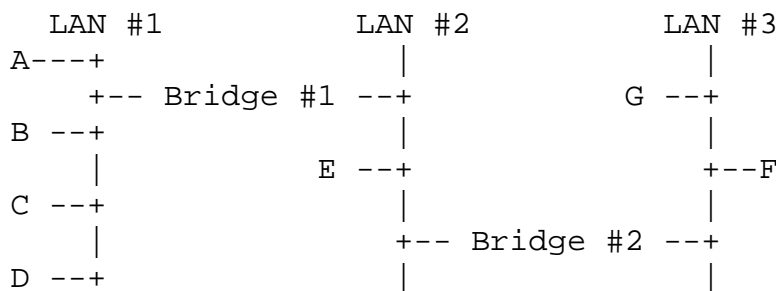
```

The variable attempts is reset to zero on a successful transmission and incremented on a collided transmission. The variable delay is the random period of time for the delay is step (3). The constant slot_time is equal to the minimum frame size (512 bits).

Problem #7

Answer the following questions about layer-2 bridging.

a) For the network below, give the contents of the bridge forwarding tables (assume that these are IEEE 802.1 transparent bridges). Assume that all stations have communicated with all other stations, EXCEPT that station A has never sent a frame (it may have received frames). Assume that the letters (A, B, C, etc.) are the station MAC addresses.



4 pts total - roughly 2 pts for each table

The table for bridge #1 is (0 = local, 1 = not local)

MAC address	Direction
B	0
C	0
D	0
E	1
F	1
G	1

The table for bridge #2 is (0 = local, 1 = not local)

MAC address	Direction (0 = local)
B	1
C	1
D	1
E	1
F	0
G	0

Station A has never sent a frame, so its source address has never been seen by a Bridge and thus MAC address A is not in any bridge table.

b) Precisely describe the Transparent Bridging (TB) learning and forwarding algorithm

4 pts total - 1 pt for each line

Each bridge runs in promiscuous mode copying all frames:

receive a frame

if (dest addr in table) and (dest addr is local) then not forward else forward

if (source addr in table) then check/update the direction

if (source addr not in table) then add it to the table

c) What is the purpose of the Spanning Tree (ST) protocol (as part of IEEE 802.1)?

2 pts total - 1 pt for each underlined item

The purpose of the ST protocol is to maintain an acyclic (tree) structure so that frames will not loop in the network. With the TB learning and forward algorithm, frames would loop forever if there were parallel bridges (i.e., loops).

Problem #8

Answer the following questions regarding wireless networks.

a) Why is Collision Detection (CD) not feasible in a wireless network?

4 pts total - 2 pts for each underlined item

In a wireless host the receiver is overwhelmed by the transmitter and cannot "hear" if someone else is transmitting. Also, hidden terminals and signal fading make it possible for two hosts A and C to not be able to hear each other, but a host B can hear both A and C. In such a situation A and C would not be able to detect a collision in any case, but B would be affected by it.

b) Describe (i.e., give) the CSMA/CA algorithm as used in WiFi (IEEE 802.11).

6 pts total - roughly 1 pt for each line

At the sender:

(1) if sense channel idle for DIFS then transmit entire frame

(2) if sense channel busy then

start random backoff time
timer counts down while channel is idle
transmit when timer expires
if no ACK received then increase the backoff time and repeat (2)

At the receiver:
if frame is received OK then send an ACK

DIFS is the Distributed Interframe Space and SIFS is the Short Interframe Space. SIFS is shorter in time than DIFS.

Problem #9

Answer the following questions regarding multimedia networking.

- a) If a new network were to be designed and built to support multimedia networking, what are the “four pillars” necessary (to use the terminology of Kurose and Ross) or the four “QoS principles” (to use the terminology used in class)?

4 pts total - 1 pt for each item

The pillars are 1) packet classification, 2) isolation of flows, 3) high network utilization, and 4) call admission.

- b) What is (or was) ATM? Explain in one of two sentences.

3 pts total - 1 pt for each underlined item

ATM is (was) Asynchronous Transfer Mode which is a virtual circuit cell-based network. ATM supported multiple traffic classes and was intended to be a universal network to carry voice, video, and data.

- c) What is (or was) RSVP? Explain in one of two sentences.

3 pts total - 1 pt for each underlined item

RSVP is a resource reservation set-up protocol for IP networks intend to support IntServ. RSVP is used by a host to request specific qualities (say, delay, loss, and throughput) from the network for particular application flows.

Problem #10

Answer the following questions regarding network management.

- a) What are the five areas of network management as identified by ISO.

4 pts total - 1 pt for each item

Performance, fault, configuration, accounting, and security.

- b) What is SNMP and how does it work? Identify the major components of a network being managed using SNMP.

4 pts total - 1 pt for each underlined item

SNMP is Simple Network Management Protocol. SNMP is a request-response protocol between a manager and an agent (with an associated MIB or data base of management variables) in a remote device being managed. The key commands are GET (a variable), SET (a settable variable), and TRAP (device sends a message when a given variable hits a threshold).

- c) What is MRTG? Explain in one of two sentences. What is an example of MRTG usage?

2 pts total - 1 pt for each underlined item

MRTG is Multi-Router Traffic Grapher and is an SNMP-based application that can GET and graph selected SNMP variables on a periodic basis. One example is to get bytes in and out counter values from routers and plot traffic utilization in and outbound on a company network. MRTG freely available (i.e., open source).

Extra Credit

How could the project experience have been improved (but, keeping the idea of the class developing a standard and then implementing it)? Give one or two specific *feasible* ideas. Also, carefully describe what you mean by “improved”. A feasible idea is one that you would have taken advantage of – that is, something you would have done (and also would not have been overbearing for the instructor and/or TA to implement).

10 pts total - Ideas that are not feasible or are trivial/silly should lose points.

Various answers are possible.

Appendix – Program tcpServer.c with some bugs

```
1. //===== file = tcpServer.c =====
2. // = A message "server" program to demonstrate sockets programming =
3. //=====
4. // = - Modified for exam #1 to include some bugs for the students to find =
5. //=====
6. //----- Include files -----
7. #include <stdio.h> // Needed for printf()
8. #include <string.h> // Needed for memcpy() and strcpy()
9. #include <windows.h> // Needed for all Winsock stuff

10. //----- Defines -----
11. #define PORT_NUM 1050 // Arbitrary port number for the server

12. //===== Main program =====
13. void main(void)
14. {
15.     #ifdef WIN
16.     WORD wVersionRequested = MAKEWORD(1,1); // Stuff for WSA functions
17.     WSADATA wsaData; // Stuff for WSA functions
18.     #endif
19.     int welcome_s; // Welcome socket descriptor
20.     struct sockaddr_in server_addr; // Server Internet address
21.     int connect_s; // Connection socket descriptor
22.     struct sockaddr_in client_addr; // Client Internet address
23.     struct in_addr client_ip_addr; // Client IP address
24.     int addr_len; // Internet address length
25.     char out_buf[4096]; // Output buffer for data
26.     char in_buf[4096]; // Input buffer for data
27.     int retcode; // Return code

28.     // Create a socket
29.     welcome_s = socket(AF_INET, SOCK_DGRAM, 0);
30.     if (welcome_s < 0)
31.     {
32.         printf("*** ERROR - socket() failed \n");
33.         exit(-1);
34.     }

35.     // Fill-in server (my) address information and bind the socket
36.     server_addr.sin_family = AF_INET;
37.     server_addr.sin_port = htons(PORT_NUM);
38.     server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
39.     retcode = bind(connect_s, (struct sockaddr *)&server_addr,
40.     sizeof(server_addr));
41.     if (retcode < 0)
42.     {
43.         printf("*** ERROR - bind() failed \n");
44.         exit(-1);
45.     }

46.     // Accept a connection.
47.     printf("Waiting for accept() to complete... \n");
48.     addr_len = sizeof(client_addr);
49.     connect_s = accept(welcome_s, (struct sockaddr *)&client_addr, &addr_len);
50.     if (connect_s < 0)
51.     {
52.         printf("*** ERROR - accept() failed \n");
53.         exit(-1);
54.     }
```

```
55. // Copy the four-byte client IP address into an IP address structure
56. memcpy(&client_ip_addr, &client_addr.sin_addr.s_addr, 2);

57. // Print an informational message that accept completed
58. printf("Accept completed (IP address of client = %s port = %d) \n",
59.     inet_ntoa(client_ip_addr), client_addr.sin_port);

60. // Send to the client using the connect socket
61. strcpy(out_buf, "This is a message from SERVER to CLIENT");
62. retcode = send(connect_s, out_buf, strlen(out_buf), 0);
63. if (retcode < 0)
64. {
65.     printf("*** ERROR - send() failed \n");
66.     exit(-1);
67. }

68. // Receive from the client using the connect socket
69. retcode = recv(connect_s, in_buf, (sizeof(in_buf) + 1), 0);
70. if (retcode < 0)
71. {
72.     printf("*** ERROR - recv() failed \n");
73.     exit(-1);
74. }
75. printf("Received from client: %s \n", in_buf);

76. // Close the welcome and connect sockets
77. retcode = closesocket(welcome_s);
78. if (retcode < 0)
79. {
80.     printf("*** ERROR - closesocket() failed \n");
81.     exit(-1);
82. }
83. }
```