Network Layer (IP)

1. Assuming that all routers and hosts are working properly and that all software in both is free of all errors, is there any chance, however small, that a packet will be delivered to the wrong destination?

   answer:

   Yes. A large noise burst could garble a packet badly. With a k-bit checksum, there is a probability of 2^{-k} that the error is undetected. If the destination field, or equivalently, "virtual circuit" number is changed, the packet will be delivered to the wrong destination and accepted as genuine. Put in other words, an occasional noise burst could change a perfectly legal packet for one destination into a perfectly legal packet for another destination.

2. The IP packet header includes a time-to-live field that is decremented by each router along the path. Why is the time-to-live field necessary?

   answer

   A packet may get stuck in a forwarding loop (e.g., due to a router configuration mistake). By decrementing the TTL field at each hop, and discarding the packet when the TTL reaches 0, the network prevents the packet from cycling in a loop indefinitely. Otherwise, the packet would consume excessive resources, or even escape the loop eventually and reach the destination much later (running the risk that the packet is mistakenly viewed as part of a more recent transmission with the same IP addresses and TCP/UDP port numbers).

3. Consider sending a 1500-byte datagram into a link that has an *Maximum Transmission* Unit of 500 bytes. Suppose the original datagram is stamped with the identification number 1. Assume that IPv4 is used. Hint: The IPv4 header is 20 bytes long.

   i. Where does fragmentation happen? Where are the fragments reassembled?

   answer

   Fragmentation happens in the router preceding the link with the small MTU. The intention is that fragments are reassembled in the end system.

   ii. How many fragments are generated?

   answer

   The maximum size of the data field in each fragment = 480 (because there are 20 bytes IP header). Thus the number of required fragments is 4

   iii. In addition to the identification number, what are the fields in the generated IP datagram(s) that are related to fragmentation?

   answer

   Flag and fragmentation offset.

   iv. What are the values of the fragmentation-related fields in the generated IP datagram(s)?

   answer

   Each fragment will have an identical identification number. Each fragment except the last one will be of size 500 bytes (including the IP header). The last datagram will be of size 60 bytes (including the IP header).

   Each of the first 3 fragments will have flag=1; the last fragment will have flag=0.

   The offsets of the 4 fragments will be 0, 60, 120, 180. Note: all fragments except the last one in a datagram must be a multiple of 8 bytes (the elementary fragment unit).

   v. What changes if IPv6 were used?

   answer

   The router preceding the link with the small MTU will drop the packet and send an ICMP error message Packet Too Big back to the source. The source is responsible for adjusting the packet

4. Suppose that instead of using 16 bits for the network part of a class B address originally, 20 bits had been used. How many class B networks would there have been?

   answer:

   With a 2-bit prefix, there would have been 18 bits left over to indicate the network. Consequently, the number of networks would have been 2^{18} or $262,144$. However, all 0s and all 1s are special, so only $262,142$ are available.

5. A network on the Internet has a subnet mask of 255.255.240.0. What is the maximum number of hosts that it can handle?

   answer:

   The mask is 20 bits long, so the network part is 20 bits. The remaining 12 bits are for the host, so 4096 host addresses exist.

6. Suppose an ISP owns the block of addresses of the form `101.101.128.0/17`. Suppose it wants to create four subnets from this block, each block having the same number of IP addresses. What are the prefixes (of form `a.b.c.d/x`) for the four subnets?

   answer

   101.101.128.0/19
   101.101.160.0/19
   101.101.192.0/19
   101.101.224.0/19

Socket Programming

1. Is the following statement true or false? With UDP sockets, a server can easily determine the IP address of the client, from the data returned via a socket read. Answer the same question (true or false?) for TCP sockets. Briefly explain your answers.

   answer

   This is true for UDP since the UDP packet contains the IP address of the sender of the UDP packet.

   See sample server udp code:

   ```
   n = recvfrom(sock,buf,1024,0,(struct sockaddr *)&from,&fromlen);

   n = sendto(sock,"Got your message\n",17, 0, (struct sockaddr *)&from,fromlen);
   ```

   This is false for a TCP socket since the socket only returns the byte stream sent by the client, but no identifying information about the client

2. What is the purpose of the connection-oriented welcoming socket, which the server uses to perform an `accept()`? Once the `accept()` is done, does the server use the welcoming socket to communicate back to the client? Explain your answer.

   answer

   The welcoming socket is used in a server to wait for incoming client connection requests in connection-oriented (TCP-based) communication. A new socket is created on return from the `accept()`. This new socket is then used by the server to communicate back to the client.

Transport Layer

1. Indicate whether TCP or UDP (or both or neither) provide the following services to applications:

   (a) Reliable data transfer between processes.

   (b) Minimum data transmission rate between processes.

   (c) Congestion-controlled data transfer between processes.

   (d) A guarantee that data will be delivered within a specified amount of time.

   (e) Preserve application-level message boundaries. That is, when a sender sends a group of bytes into a socket via a single send operation, that group of bytes will be delivered as a group in a single receive operation at the receiving application.

   (f) Guaranteed in-order delivery of data to the receiver.

answer

(a) TCP provides a reliable byte-stream between client and server.

(b) Neither

(c) TCP

(d) Neither

(e) UDP preserves message boundaries, while TCP is byte-stream oriented, and does not preserve message boundaries.

(f) TCP will deliver bytes to the receiver in the order in which they were sent. UDP does not guarantee delivery of message data in the order in which they were sent.

2. Suppose you use UDP to do a transaction from a remote client to a server. UDP provides no reliability, but you want your transaction request to be sent reliably. How could you do is?

   answer

   A little ambiguous – You would build reliability into your application. This could be done, for example, by having the client re-transmit its request if it doesnt hear back from the server within a certain amount of time.

3. Why does UDP exist? Would it not have been enough to just let the user processes send raw IP packets?

   answer:

   No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process.

4. Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Given two reasons for why these protocols invented a new abstract ID (port numbers), instead of using process IDs, which already existed when these protocols were designed?

   answer:

   Here are three reasons. First, process IDs are OS-specific. Using process IDs would have made these protocols OS-dependent. Second, a single process may establish multiple channels of communications. A single process ID (per process) as the destination identifier cannot be used to distinguish between these channels. Third, having processes listen on well known ports is easy, but well-known process IDs are impossible.

5. TCP provides reliable end-to-end data transmission over an unreliable network layer. What do the terms *reliable* and *unreliable* mean in this context? In your answer, you should clearly differentiate between the functionality provided in each of the corresponding layers in the protocol hierarchy.

   answer:

   see lecture notes

Transport Layer

1. What is meant by a *handshaking protocol*? Use an example to illustrate your answer.

   answer:

   When computer1 connects to computer2, the first thing it has to do is contact computer2 and tell it who it is and that it wants to connect. Computer2 then must decide to accept (or deny or ignore) the connection and then tell Computer1 that the connection is accepted.

   The protocol is just the series of technical rules that this basic process must follow. For example, when a tcp connection is initiated by computer1 the protocol is to send a SYN packet to a specific port on Computer2. Computer2 then, if accepting the connection, sends an ACK packet back to Computer1. From that point on the tcp connection is considered open and all future packets will have both the SYN and ACK bits until the connection is closed.

2. Review the fields in the TCP, UDP, and IP headers. Briefly describe the role of the "important fields" from each of the headers. (I know "important fields" is ambiguous .... but my objective is to get you to think about the Wireshark tasks from the lab).

   answer:

   Your answer should include a discussion of:

   TCP = port numbers, seq/ack number, flags, checksum, and windows size

   UDP = port numbers and checksum

   IP = addresses, checksum for header

   Note: you will not be asked such an ambiguous question in the exam.

3. **A multiple choice question:**

   Host A sends a TCP segment (Seq = 43, ACK = 103), to which host B replies with a TCP segment (Seq = 103, ACK = 57). The payload of the first TCP segment is

   (a) 14 bytes long

   (b) 43 bytes long

   (c) 46 bytes long

   (d) 57 bytes long

   (e) 60 bytes long

   answer:

   (a)

4. Datagram fragmentation and reassembly are handled by IP and are invisible to TCP. Does this mean the TCP does not have to worry about data arriving in the wrong order? Justify your answer.

   answer:

   Even though each datagram arrives intact, it is possible that the datagrams arrive in the wrong order, so TCP has to be prepared to reassemble the parts of the message properly.

5. The maximum payload of a TCP segment is 65, 495 bytes. Why was such a strange number chosen?

   answer:

   The entire TCP segment must fit in the 65, 515-byte payload field of an IP packet. Since the TCP header is a minimum of 20 bytes, only 65, 495 bytes are left for TCP data.

6. TCP and UDP provide two very different service models. Suppose that an application wants all of the functionality provided by UDP but only some of the functionality provided by TCP (e.g., the application wants reliable message transfer and flow control, but not congestion control). How would an application get this different service in todays Internet?

   answer:

   Answer: The application would use UDP sockets and implement the desired additional functionality (e.g., reliability and flow control) in the application itself.

7. Explain the concepts of slow start, fast retransmit and fast recovery in TCP and their effects on TCP performance.

Network Layer

1. The IP packet header includes a time-to-live field that is decremented by each router along the path. Why is the time-to-live field necessary?

   answer

   A packet may get stuck in a forwarding loop (e.g., due to a router configuration mistake). By decrementing the TTL field at each hop, and discarding the packet when the TTL reaches 0, the network prevents the packet from cycling in a loop indefinitely. Otherwise, the packet would consume excessive resources, or even escape the loop eventually and reach the destination much later (running the risk that the packet is mistakenly viewed as part of a more recent transmission with the same IP addresses and TCP/UDP port numbers).

2. A router has the following entries in its routing table:

Address/mask	Next hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Router 1
default	Router 2

   For each of the following IP addresses, what does the router do if a packet with that address arrives? a) 135.46.63.10, b) 135.46.57.14, c) 135.46.52.2, d) 192.53.40.7, e) 192.53.56.7

   answer:

   The packets are routed as follows: a) Interface 1, b) Interface 0, c) Router 2, d) Router 1, e) Router 2

3. List one motivation for a host to send an IP packet with the wrong source IP address. List two ways that this can adversely affect the legitimate owner of that IP address.

   answer

   A host launching a denial-of-service attack may send packets with a spoofed source address that corresponds to another host, in order to evade detection. The legitimate owner may be blamed for the attack (and perhaps also blocked from sending legitimate traffic to the victim destination), and may also receive unwanted return traffic (e.g., SYN-ACK or RST packets).

Application Layer

1. Does a DNS server for a domain have to be on the same network as the hosts whose names it resolves?

   answer:

   No, the server does not need to be on the same network because local DNS servers can forward the queries to other remote servers.

2. DNS uses UDP instead of TCP. If a DNS packet is lost, there is no automatic recovery. Does this cause a problem, and if so, how is it resolved?

   answer:

   DNS is idenmpotent. Operations can be repeated without harm. When a process makes a DNS request, it starts a timer. If the timer expires, it just makes the request again. No harm is done.

3. Describe the basic operation of a Web server using high-level pseudocode. It is sufficient to show this basic operation for only the case of HTTP GET.

   answer

   see lecture notes

4. The standard HTTP URL assumes that the Web server is listening on port 80. However, it is possible for a Web server to listen on some other port. Devise a reasonable syntax for a URL accessing a file on a nonstandard port.

   answer:

   The official RFC 1738 way to do this is *http://dns-name:port/file.*

5. Many Web browsers open several TCP connections in parallel when downloading multiple embed-ded images. How does this affect the other TCP traffic sharing the same congested link?

   answer

   The extra TCP connections allow the Web browser to have an unfair share of the bandwidth of the bottleneck link, reducing the throughput of other, well-behaved applications.

   When no other traffic traverses the link, the multiple connections share bandwidth only with each other. Still, this may be beneficial if the connections are RTT-limited, and because the user likes to see multiple embedded images fill in at the same time; also, the parallel connections collectively get more bandwidth during slow start.

6. Web caches are often justified on the ground that they speed up web browsing and reduce band-width costs, but sometimes they do not work well. Provide reasons why a bad web cache might not be a good investment for an organization.

   answer

   This answer was discussed in the lecture – also, see discussion in Kursoe and Ross