भारतीय प्रौद्योगिकी संस्थान खड़गपुर
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Department of Computer Science and Engineering

CS 31006: Computer Networks **Class Test – 1**
Full Marks: 25 Time: 45 Minutes Spring, 2018

1. State whether the following statements are TRUE or FALSE. Write a justification (within maximum 50 words) against your answer. **Note: Marks will only be given if both the TRUE/FALSE assignment as well as the justification are correct – there is no separate marks for justification only**.

   (a) All the devices in the network need to support all five layers of the protocol stack.
   Answer: FALSE
   Only the end hosts in a network are required to support all five layers of the protocol stack; intermediate devices may support only upto a certain layer. Which layers a certain type of device will support, is decided based upon its functionality in the network. For example, a link-layer switch is responsible for transfer of frames to a device in its direct vicinity, so it is sufficient for it to have knowledge of ethernet addresses (implements till layer 2). In contrast, a router is responsible for transferring datagrams using IP routing, which requires that it stores a routing table (and thus implements till layer 3).

   (b) A single protocol instance at the transport layer can handle data from multiple applications.
   Answer: TRUE
   The same transport layer instance uses port numbers to handle data from multiple applications. It uses multiplexing and demultiplexing to deliver data from the source process (socket) to the destination process (socket).

   (c) HTTP, by default, maintains the state of the client while sending a HTTP Request Message.
   Answer: FALSE
   HTTP is a stateless protocol by default. If required, cookies are used to store state information.

   (d) The three-way handshaking mechanism used for connection release of a connection oriented transport protocol always ensures **no data loss** from that connection.
   Answer: FALSE
   Although the three-way handshaking mechanism is usually adequate in practice, it does not guarantee that no data will ever be lost. The two-army problem helps to illustrate this point.

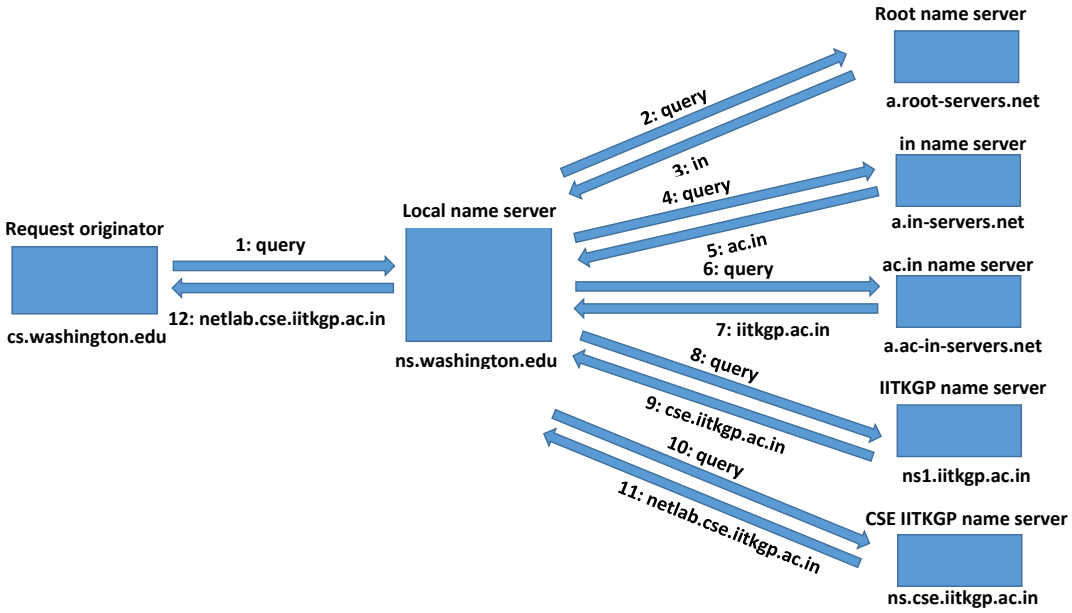
   (e) One bit sequence number is sufficient for stop-and-wait ARQ.
   Answer: TRUE
   One bit sequence number is sufficient since the sender waits for acknowledgment of receipt of a 0-frame, before starting transmission of a 1-frame, and so on. In case a frame gets lost, it is expected to die out before another frame with the same sequence number is generated.

[2x5]

2. Answer the following questions. The answers should be precise and to the point. Unnecessary write-ups may result in penalties in the form af additional marks deduction.

   (a) Explain, with a diagram, how DNS resolves a query (finds out the IP address) for the URL net-lab.cse.iitkgp.ac.in, when you access it from a network which is completely outside the network where the system corresponds to netlab.cse.iitkgp.ac.in resides. You can assume that the name server for cse.iitkgp.ac.in maintains the resource record for netlab.cse.iitkgp.ac.in.

   Answer: Let us assume that the DNS query has been generated from the domain `cs.washington.edu`. The following figure shows the progression of requests and responses in the DNS resolution process:



   [3]

   (b) HTTP and FTP, both are primarily used to fetch files from a server machine. Then, why do we choose two different protocol primitives for these two protocols? For instance, HTTP uses a single connection at port 80, whereas FTP uses two different connections – one at port 21 for control/command messages and another port for data transfer.

   Answer: FTP uses 2 ports in order to avoid busy waiting, and to keep the command channel lightweight. While multiplexing between control and data (such as in the case of HTTP) is possible, in that case, while one client is being served, others would face significantly higher queuing delay (since FTP is often used to transfer large files). With separate ports in operation, multiple clients can continue sending and receiving control information, while data transfer is being take place. [3]

   (c) Why does most of the DNS implementations prefer UDP messages for query resolution over a name server?

   Answer: DNS implementations generally prefer UDP over TCP, since UDP is faster. TCP consumes time during handshaking; since DNS uses a cascading approach for name resolution, with TCP, for every such message, separate connection setups would be required. Another factor is that DNS requests and responses are generally very small, and fit well within one UDP segment.

While UDP is not reliable, reliability in case of DNS is ensured at the application layer, using timeouts. [3]

(d) What is the purpose of connection establishment at the transport layer?

Answer: The transport layer deals with creating and monitoring a logical pipe between the sender and the receiver. It needs to know the state of the pipe at every moment, in order to take appropriate actions. Connection establishment ensures that both ends of the pipe are always aware of the current state of the pipe (thus making the transport layer protocol a stateful protocol). [3]

(e) Why do we use two different protocols (SMTP and POP/IMAP) for email transfers (one for sending emails and another for fetching emails)?

Answer: SMTP is primarily a *push* protocol – the sending mail server pushes the file to the receiving mail server. However, a separate *pull* protocol is required (e.g., POP/IMAP), in order to enable users to read emails by executing a mail client, on an end device with the ability to view multimedia messages and attachments. While one protocol for both functionalities is possible, the performance overhead dictates that send and receive are performed using two separate email protocols. [3]

भारतीय प्रौद्योगिकी संस्थान खड़गपुर
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Department of Computer Science and Engineering

CS 31006: Computer Networks **Mid Semester Examination**
Full Marks: 50 Time: 2 hours Spring, 2018

1. State whether the following statements are TRUE or FALSE. Write a justification (within maximum 50 words) against your answer. **Note: Marks will only be given if both the TRUE/FALSE assignment as well as the justification are correct – there is no separate marks for justification only**.

   (a) Flow control at the transport layer is not essential, but sufficient. (Hint: By *essential*, we indicate that the property X needs to be there at layer Y for correct operations of the protocol; by *sufficiency*, we indicate that if we implement property X at layer Y, and we do not need to implement it at any other layer of the protocol stack.)

   Answer: FALSE

   Flow control is essential at the transport layer in order to ensure that the sender does not overwhelm the receiver. However, it is not sufficient at the transport layer, and is required at the data link layer as well. While the transport layer is concerned with end-to-end flow control, the link layer ensures flow control at a node-to-adjacent-node basis.

   (b) Using seperate control/command channel and data channel in FTP helps in reducing the average response time for individual FTP clients.

   Answer: TRUE

   FTP is often used to transfer large files. If the same channel were used for control and command, while one client is being served, others would have to wait. With separate ports, multiple clients can continue sending and receiving control information, along with data transfer, thereby reducing the average response time.

   (c) Stop and Wait flow control requires a protocol to be connection oriented.

   Answer: TRUE

   A connection-oriented protocol maintains state information of the sender and the receiver. Since Stop and Wait flow control relies on acknowledgments of packets sent, it requires a connection-oriented protocol.

   (d) A persistent connection is **essential** for loading dynamic web pages (web pages with client side scripts that dynamically load the content based on user input).

   Answer: FALSE

   A persistent connection is not essential; different components in a dynamic web page may be downloaded over multiple connections. This results in higher overhead, but is always possible.

(e) DNS resource records are **always consistent** across all the name servers.

Answer: FALSE

DNS records are not necessarily always consistent across all name servers. For example, if the IP address mapped to a hostname changes, the change may not reflect immediately across all name servers in the hierarchy. The authoritative records are the reliable ones in such cases.

(f) FTP uses TCP as the transport layer protocol. An FTP server *always needs* to run the corresponding TCP server for data transfer at the transport layer.

Answer: FALSE

If operating in the *active* mode, the client starts listening on a port of its choice, and communicates the same to the server, which then establishes the data channel. The reverse happens in the *passive* mode; the server starts listening on a port of its choice, and the client connects to that port, after obtaining information about it.

(g) It is **essential** to start the TCP slow start with congestion window size of 1.

Answer: FALSE

It is not essential to set the congestion window size at 1, it is more of a convention. A small value of the window usually suffices.

(h) Assume a network with bandwidth 1 Mbps and RTT 1 sec. Then a sliding window protocol with window size of 512 Kb gives 100% link utilization.

Answer: FALSE

For 100% link utilization, the window size should be equal to the amount of data that has left the sender, until it has received the first ACK. This is equal to the BDP (defined w.r.t. RTT, not w.r.t. one-way delay).
BDP, in this case = $1\ Mbps\ *\ 1\ sec = 1\ Mb = 1024\ Kb$
However, the window size here is $512\ Kb$, which is lesser than required for max. utilization.

(i) Additive Increase Additive Decrease (AIAD) approach for adapting the sending rate for congestion control can help in achieving 100% efficiency of a transport protocol.

Answer: TRUE

Both AIAD and MIMD can potentially achieve 100% efficiency (they oscillate across the efficiency line, with different slopes). However, since fairness is another desired property of a transport protocol, AIMD is preferred, since it converges towards optimal performance (w.r.t. both efficiency and fairness).

(j) The message transfer agents need to run the SMTP server all the time at a well known port. [2x10]

Answer: TRUE

SMTP servers are always available in the background. The originating user agent establishes a TCP connection to the SMTP server on its well-known port (25 by default), and sends across a new email.

2. Answer the following questions. The answers should be precise and to the point (within 50 words each). Unnecessary write-ups may result in penalties in the form af additional marks deduction.

(a) Assume that your email address is cs31006@cse.iitkgp.ac.in. From this email address, you want to send an email to sandipc@iitg.ernet.in. Then the message transfer agent (SMTP server) for cse.iitkgp.ac.in needs to find out the SMTP server IP for the message transfer agent corresponding to the email domain iitg.ernet.in. Explain, how this name resolution for message transfer

agent is done with the help of DNS. Which particular resource entry in a DNS resource record helps in this name resolution? [2]

Answer: The SMTP server contacts the local name server (say ns1.cse.iitkgp.ac.in) with a request to resolve the recipient SMTP server. The local NS sends a DNS request to the root NS for *in* (say a.root-servers.net) and receives the address corresponding to *ernet.in*. Subsequently, it sends a DNS request to the root NS for *ernet* (say a.ernet-servers.net), and receives the address corresponding to the domain *iitg.ernet.in*.

The MX (mail exchanger) resource entry in a DNS records helps in email server resolution. In case an MX record is absent, the A record is used as a fallback.

(b) Explain an advantage of using persistent HTTP connections over non-persistent HTTP connections. Is there any scenario, when you'll **prefer to use a non-persistent connection**. [2]

Answer: A persistent HTTP connection allows for *connection reuse*, where multiple requests and responses can be handled in one connection. This enables *pipelined requests*, unlike in non-persistent HTTP, thus reducing latency and CPU overhead.

In cases where only one request-response cycle is sufficient to serve the purpose of the user, a persistent connection may continue to hold resources, which would be made available to other connections in the non-persistent case.

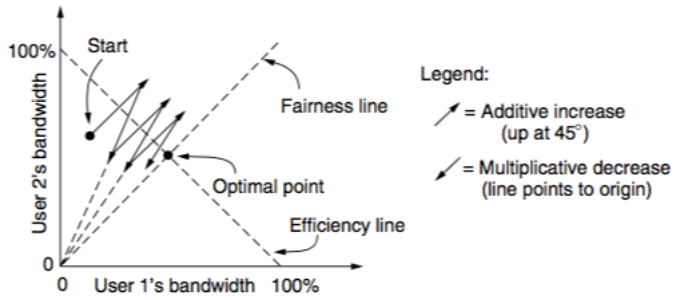
(c) Why TCP does not close a connection immediately after sending a FIN message or receiving the ACK corresponding to a FIN message (it goes to a TIME_WAIT state)? [2]

Answer: (1) If the sender closes the connection immediately after sending a FIN, this FIN may get lost and may cause the receiver to wait indefinitely for more data.
(2) If the FIN is received at the receiver, and the corresponding FIN+ACK is sent by it, it still doesn't know whether the sender is aware of its willingness to terminate. The sender sends an ACK to confirm this.
The two-army problem illustrates that there is no way to absolutely ensure consensus (since the final ACK may always be lost).

TCP attempts to solve this problem by entering the TIME_WAIT state, where it waits for sufficient amount of time to ensure that all packets in the connection have died off; if the receiver hasn't resent FIN+ACK by this time, it is presumed to have received the final ACK.

(d) With the help of a diagram, explain how Additive Increase Multiplicative Decrease (AIMD) helps in achieving fairness? [2]

Answer: AIMD converges towards the optimal point between efficiency and fairness, as illustrated by the following figure.



(e) Assume a TCP implementation, where the Nagle's algorithm is implemented at the sender side and delayed acknowledgment mechanism is implemented at the receiver side. Do you see any problem in this implementation? Explain. (Only Yes/No does not carry any marks!) [2]

Answer: Yes, such an arrangement will lead to problems. Since the sender uses Nagle's algorithm, it sends the first segment and starts buffering subsequent data until the ACK to this segment is received. Now, since the receiver employs delayed acknowledgment, it waits for some time (up to 500 msec), before sending an acknowledgment, in the hope that more data will be received. This will result in a state of starvation.

(f) We already have the congestion control algorithm in the transport layer that adjusts the sending rate dynamically based on the observation of flow performance (packet loss or timeout). Why do we still need flow control at the transport layer? [2]

Answer: Congestion control is concerned with overall performance and fair allocation of resources among connections in the network. However, it does not specifically address the scenario where a receiver (in a TCP connection) is overwhelmed by a fast sender. Flow control caters to this requirement, by allowing the receiver to control the pace of the sender.

(g) Assume a transport layer protocol where *segment sizes are dynamic*, but the *connections are not loaded* (number of segments transmitted per second is low). Which of the following does give a better implementation of the connection buffer in terms of memory utilization – (i) chained variable sized buffer (A linked list of buffers of different sizes), (ii) one large circular buffer (say, a circular queue)? [2]

Answer: Case (i) should provide better memory utilization. Since load is low, a small amount of memory is expected to be used at any point in time. However, this quantity is dynamic. A large circular buffer would occupy a large amount of memory at all times, irrespective of the requirement. A linked list implementation ensures that memory is allocated only when needed.

(h) Consider the three-way handshake mechanism with sequence number negotiation for the transport layer connection management. Explain how a host will react if both the connection request and the acknowledgment are delayed duplicates of the original connection request and the original acknowledgment to that connection request. [2]

Answer: Say host A initiated a connection to host B. The original connection request from A to B, which is now delayed, proposed using sequence number x for traffic from A to B. Host B has no idea it is delayed, and replies proposing sequence number y for traffic from B to A. What it gets in return, is an old ACK from A, which acknowledges sequence number z instead, for traffic from B to A. B thus realizes that the CR is old, and drops the connection.

Host A, on the other hand, receives and ACK for it's old CR with sequence number y. It realizes that it did not send a CR to B recently, and sends back a REJECT for sequence number y.

(i) Assume that you have set up a TCP connection over a **lossless link** with end-to-end bandwidth 2 Gbps. Further assume that you are using a 16 bit byte sequence number. If the end to end link delay is 50 msec, is it safe to use a 16 bit sequence number field for a sliding window based flow control algorithm? [Hint: By safe, we indicate that the protocol will be able to distinguish between different segments. Further note that we are using simple sliding window and not an ARQ, as the link is lossless.] [2]

Answer: No, it's not safe.
The wrap-around time = $\frac{Max.\ no.\ of\ bits\ handled\ by\ a\ 16-bit\ seq.\ no.}{Bandwidth\ in\ bps}$

$= \frac{2^{16}*8}{2*2^{30}} = \frac{2^{19}}{2^{31}} = 2^{-12}$ sec.
Now, delay = 50 msec = $\frac{1}{20} \approx 2^{-4}$ sec.
The delay is much larger than the wrap-around time, which means that there may be repetitions of the seq. nos.

(j) TCP estimates RTT based on the time difference between the time when a segment is sent and the time when the corresponding acknowledgment is received. Do you see any problem of this mechanism if you use the TCP connection over a link with high packet error rate? [2]

Answer: If the error rate is high, segments and their corresponding ACKs will often be received in invalid states, triggering retransmissions. For example, the sender may receive a garbled segment, which may have been an ACK to the previous segment sent by it. The receiver resends this ACK, and is now received in its proper condition by the sender. This erroneously adds to the total time noted by the sender as RTT (here, it becomes 3*one-way delay + receiver timeout).

3. Consider a transport layer protocol, where Go-Back-N ARQ is used to ensure flow control and reliability.

(a) You have two different ways of implementing the acknowledgment (ACK) mechanism – (i) the transport layer entity of the receiver sends an ACK immediately after the segment is received at the receiver buffer, and (ii) the transport entity sends an ACK only when the application reads the data from the buffer.

Explain the relative merits and demerits of the above two mechanisms. [2+2]

Answer: In case (i), the application may read segments much slower than they arrive at the transport layer. This would cause a lot of packets to buffer, eventually exhausting it. However, this implementation does not impede the sender from sending packets fast, resulting in better bandwidth utilization.
In case (ii), the ACK mechanism is the true representation of the application behaviour, and prevents the receiver from getting overwhelmed by a fast sender. However, bandwidth utilization is low in this case, since the sliding window proceeds slower.

(b) Assume that the protocol uses byte sequence number, and the sequence number field is 16 bits. What would be the maximum size of the sender window? Assume that this protocol is used to transfer data over a link of 600 Mbps with round trip propagation delay as 50 msec. What is the percentage of time that the sender will remain waiting for the acknowledgments? [2+2]

Answer: Max. size of the sender window = Max. sequence no. = $2^{16} - 1$ = 65535 bytes.
Time required to send one full window $\approx \frac{8*2^{16}}{600*2^{20}}$ sec = $\frac{1000}{1200}$ msec = 0.83 msec. Time spent waiting for acknowledgments = RTT - time spent sending = $50 - 0.83 = 49.17$ msec. Therefore, percentage of time spent waiting for acknowledgements = $\frac{49.17}{50} * 100\% = 98.34\%$.

(c) Explain how this protocol will react if there is a loss in the acknowledgment. [2]

Answer: If the ACK from the receiver is lost for a window, there is a timeout at the sender, The sender resends the segments in its current window. Once they reach the receiver, it discards the segments (since it already has them), and retransmits the ACK.

भारतीय प्रौद्योगिकी संस्थान खड़गपुर
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Department of Computer Science and Engineering

CS 31006: Computer Networks **Class Test – 2**
Full Marks: 25 Time: 1 Hour Spring, 2018

1. Answer the following questions. The answers should be precise and to the point. Unnecessary write-ups may result in penalties in the form af additional marks deduction.

   (a) Consider that a IPv4 subnet has the subnet mask 255.255.255.224. What is the maximum number of IPv4 addresses that can be assigned to hosts on this subnet?

   Answer: The subnet mask 255.255.255.224 is equivalent to the subnet mask $/27$, which leaves room for $2^{32-27} = 2^5 = 32$ hosts. However, since all zeros (network address), and all ones (broadcast address) are not allowed, a max. of $32 - 2 = 30$ IPv4 addresses can be assigned.

   (b) Why do we keep both the next hop IP address and the default interface to reach that IP address as a part of a CIDR routing table?

   Answer: A router typically has multiple physical interfaces, with each connected to (having an IP address from) a different subnet. Therefore, in order to forward a packet, the router requires not only the next hop IP, but also the physical interface along which this IP can be reached. Otherwise, it would have to broadcast ARP requests along all interfaces, resulting in significant overhead.

   (c) Why does BGP maintain the entire path information as a sequence of autonomous systems, rather than just keeping the immediate next autonomous system?

   Answer: BGP enables routing among autonomous systems (ASes) with different business policies. Say, AS2 has a path to AS5 via both AS3 and AS4. AS1 can reach AS5 via AS2 (and then one of AS3 and AS4). However, policy at AS1 does not allow traffic to be forwarded through AS3. This policy information needs to be propagated to AS2 in order to enforce it, (by avoiding AS3, and choosing AS4). Additionally, maintaining entire path information enables easy detection of routing loops.

   (d) Explain, how distance vector routing mechanism can prevent routing loops. You may consider a general scenario when there is no link failure.

   Answer: The *split horizon* method can be used to prevent routing loops in this scenario. According to this technique, routing information is prevented from exiting the router on an interface, through which the information was received.

   (e) Assume that an institute got an IPv4 address pool 202.141.176.0/21, and the institute administrator creates a subnet 202.141.182.0/23 from that address pool. Do you see any problem in this subnet? Explain your answer.

   Answer: The administrator has created a *all-ones* subnet. This means that the broadcast address for the original subnet, and the one created by the admin, have become the same, which is the problem in this case. [2x5]

2. Alexa Pvt. Ltd. is a multi-tier company that has 4 different departments spanned over 4 floors of their office. The company makes an estimate that the 4 departments will have 100, 200, 300 and 400 different computers with a single network interface card. All the computers require public IPv4 address. Accordingly, the company has requested for an IP pool for 1024 different IPv4 addresses from the IRINN. Accordingly, IRINN has allocated the CIDR IPv4 address pool 202.110.180.0/22 to Alexa.

   (a) After a couple of months, Alexa recruits Dr. Echols as their network administrator. Dr. Echols suggests to have four different CIDR subnets for the four different departments. Can she construct four different subnets from the allocated IP address pool? If she can, then design a possible allocation, otherwise explain why it is not possible. [5]

   Answer: In the original address pool, 10 bits are allowed for allocation to computers in the company. If separate subnets are created, the departments with 300 and 400 machines will require 9 bits each for addressing (with 8 bits, only $2^8 - 2 = 254$ addresses are possible; with 9 bits, $2^9 - 2 = 510$ addresses are possible). Allocating 9 bits for a subnet leaves only 1 bit for the subnet address, which can either be set to 0 or 1. Since *all-zero* and *all-one* subnets are not allowed, this allocation is not possible.

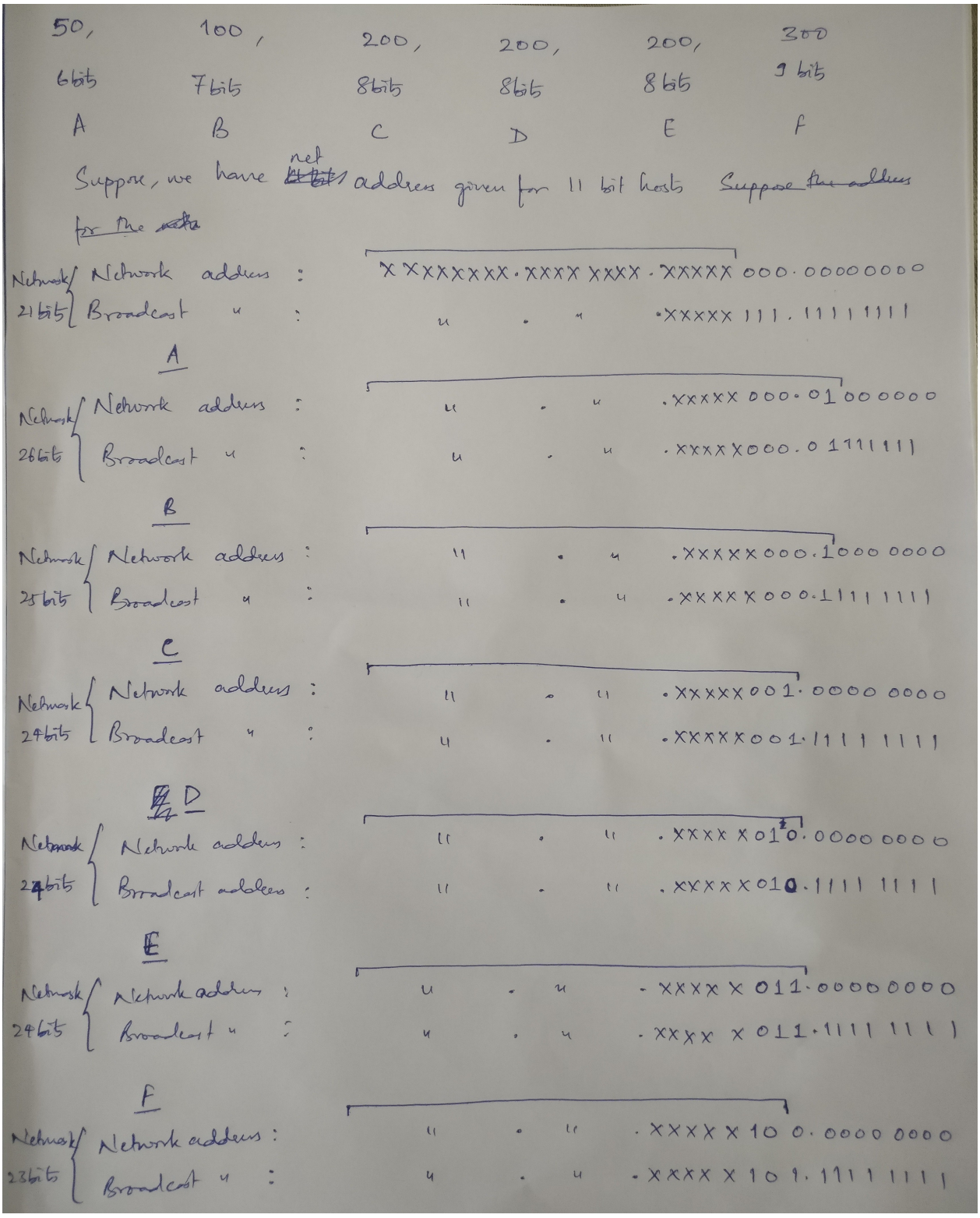
   (b) Assume that after 2 years, the fourth department got broken into two departments of 200 computers each, and another new department got added with 50 new computers with a single network interface card. So, now there are six different departments, with 100, 200, 300, 200, 200 and 50 machines, respectively. At this point, Dr. Echols wants to have six different subnets for the six departments. How many minimum number of additional CIDR IPv4 address now she should request for, so that she can construct the subnets safely with all public IPv4 addresses? Note that Dr. Echols is a methodical engineer, and so she wants to avoid all one and all zero subnets. [10]

   Answer:

   Note: The most optimal solution (*Solution 1*), which was offered by Nishant, requires 11 bits, and therefore requires only 1024 extra IPv4 addresses. However, there can be sub-optimal solutions with 12 bits, and we explain one of those solutions (*Solution 2*) too.

   Marking Scheme: A solution with 11 bits will be evaluated out of 15 marks for this answer, while a solution with 12 bits will be evaluated out of the originally offered 10 marks. The marks will be normalized later during grading.

Solution 1 (optimal solution – thanks to Nishant): Only 11 bits are required when this solution is used, requiring $2^{11} - 2^{10} = 1024$ extra IPv4 addresses.



Solution 2 (sub-optimal solution):

In order to accommodate 300 machines in one subnet, 9 bits are required for addressing them (with 8 bits, $2^8 - 2 = 254$ addresses are possible, while with 9 bits, $2^9 - 2 = 510$ addresses are possible). Avoiding all-zero and all-one cases, 3 bits would be required for the subnet addresses (namely, 001, 010, 011, 100, 101, and 110). Therefore, 12 bits are required in total for host addresses, so that Dr. Echols can create the 6 different subnets. In other words, she requires $2^{12} - 2^{10} = 3072$ new IPv4 addresses.

भारतीय प्रौद्योगिकी संस्थान खड़गपुर
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Department of Computer Science and Engineering

CS 31006: Computer Networks **End Semester Examination**
Full Marks: 60 Time: 3 hours Spring, 2018

Note:

(i) There are FOUR questions in this paper. Answer all questions. The answers should be precise and to-the-point.

(ii) Write down the assumptions clearly, if any. No queries will be entertained during the exam hours.

1. (a) Why it is preferred to use UDP based transport protocol for DNS? State one advantage and one disadvantage of DNS caching. [2+2]

Answer: DNS implementations generally prefer UDP over TCP, since UDP is faster. TCP consumes time during handshaking; since DNS uses a cascading approach for name resolution, with TCP, for every such message, separate connection setups would be required. Another factor is that DNS requests and responses are generally very small, and fit well within one UDP segment. While UDP is not reliable, reliability in case of DNS is ensured at the application layer, using timeouts.

Advantage: DNS caching reduces the time required to serve a DNS request, since it reuses the response to a previous DNS query for the same domain.

Disadvantage: If the address of a domain changes, a DNS serve may end up sending a stale response, until the TTL for that DNS record expires; this results in browsing issues for the end user.

(b) HTTP 1.0 marks the end of a request-reply by closing the underlying TCP connection. Explain, in terms of TCP, why this can be a problem for dynamic web pages. [2]

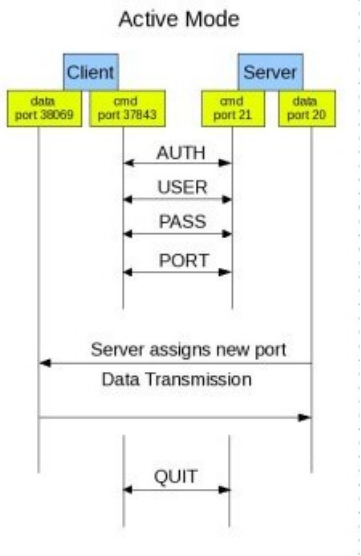
Answer: Dynamic web pages typically request for multiple webpage components (e.g., image, form response, etc.) based on user interaction. Multiple components would trigger multiple request-reply cycles, which in turn would create a new TCP connection for each such cycle. In order to establish a new TCP connection, 3-way handshaking needs to take place before useful data is transferred. This incurs large overhead, and is therefore a problem.

(c) Why do e-mail servers use two different protocols – one for email transfer and another for email receive? [2]

Answer: During email transfer, a *push*-type protocol (e.g. SMTP) is required, using which the sending mail server pushes the email to the receiving mail server, via message transfer agents. However, a separate *pull*-type protocol is required (e.g., POP/IMAP), in order to enable users to read emails by executing a mail client, on an end device with the ability to view multimedia messages and attachments. While one protocol for both functionalities is possible, the performance overhead dictates that send and receive are performed using two separate email protocols.

(d) Explain, with a diagram, how TCP connections are initiated in active FTP. Why this can be a problem for certain cases. [2+2]

Answer: The connections for FTP active-mode are shown in the following diagram:



In the active mode, the FTP server initiates a TCP connection to a port advertised by the client. However, if the client is behind a firewall, the server cannot initiate a connection to the client, which will cause problems.

(e) Assume that a machine has a single DNS name. Can this machine have multiple IP addresses? Explain how the name resolution might work in this case. [3]

Answer: Yes, it is possible for a machine with a single DNS name to have multiple IP addresses. Such a strategy is used for DNS load balancing, where a DNS query is resolved by sending one IP address from a pool of associated IP addresses.

2. (a) Consider a 1 Gbps link with 10 ms end-to-end one way delay. How much time will it take to wrap around the sequence numbers if the sequence number field is 32 bits long? [2]

Answer: Wrap-around time $= \frac{Max.\ no.\ of\ bits\ handled\ by\ a\ 32-bit\ seq.\ no.}{Bandwidth\ in\ bps} = \frac{2^{32}*8}{2^{30}} = 32$ sec.

(b) Assume that a host has received two SYN packets from the port from another remote host. There can be two reasons for this – either (i) the second SYN is a retransmission or (ii) the remote host has crashed and the second SYN is a new SYN after the remote host rebooted. How can the host differentiate between these two cases? [3]

Answer: In case of TCP, the host only remembers sequence nos. of packets which are present in its current window. Therefore, it cannot directly distinguish between cases (i) and (ii). When it sends an ACK to the connection initiator, the initiator can then differentiate according to its sequence no., and proceed accordingly.

Note: Many students have answered that the host will differentiate using the sequence no. of the SYN packet. While this is not correct for TCP, marks have been awarded for this solution also.

(c) Explain why RTT estimation is important for TCP adaptive retransmission. Why the variance of RTT is also considered for RTT estimation in Jacobson's algorithm. [2+2]

Answer: Continuous RTT estimation is important for TCP adaptive retransmission because the RTT in case of transport layer is highly variable and stale estimates may not correctly represent the current conditions.

Jacobson's algorithm considers variance of RTT because the RTT fluctuates heavily in certain cases, especially at high load. A constant value would be too inflexible to deal with such fluctuations.

(d) TCP may trigger a congestion control for two different signals – (i) a timeout and (ii) reception of three duplicate acknowledgments. Explain how these two signals are different in terms of indicating the severity of the congestion in the network. Is it necessary to set CWnd to 1 MSS if a congestion is detected through three duplicate acknowledgments? Explain. [3]

Answer: (i) Timeout is a sure sign of congestion, and indicates more severity. (ii) 3 duplicate ACKs is a softer way of determining congestion, since duplicate ACKs mean that some segments are still getting transferred in the network. Dup-ACKs thus usually indicate temporary congestion. The advantage is that the 3 dup-ACKs strategy takes lesser time to indicate congestion than a timeout.

It is not necessary to set CWnd to 1 MSS; e.g., TCP Reno implements *fast recovery*, where the CWnd is set to ssthresh+3, with ssthresh first being set of $\frac{1}{2}$ of current CWnd.

(e) Explain how fast recovery can help in early retransmission of lost segments. [3]

Answer: In fast recovery, the CWnd is set to ssthresh+3, when 3 dup-ACKs have been received. Then, each time a dup-ACK arrives, CWnd is incremented by 1. A new data segment is then sent if CWnd allows it. When a new ACK, which acknowledges all packets between lost packet and first dup-ACK, is received, the fast recovery phase is exited. Fast recovery thus helps in early retransmission of the lost segments, by inflating the CWnd, in sharp contrast to the case where fast recovery is not used, and CWnd is set to 1 MSS.

3. (a) Consider the following routing table with IPv4 CIDR notation (a simplified version).

Subnet IP	Subnet Mask	Next Hop	Default Interface
128.91.39.0	255.255.255.128	202.141.81.2	eth0
128.91.39.128	255.255.255.128	202.142.80.1	eth1
128.91.40.0	255.255.255.128	202.140.81.1	eth2
191.2.153.0	255.255.255.192	128.80.2.2	eth3
0.0.0.0	0.0.0.0	128.80.1.1	eth4

Find out the interface where the IPv4 packets will be forwarded, where the destination IPv4 address is given as follows.

i) 128.91.39.11
ii) 128.91.40.14
iii) 128.91.40.151
iv) 191.2.153.18
v) 192.2.153.92

Give one line reason for each case. [10]

Answer:
(i) eth0 – 128.91.39.11 belongs to the subnet with IP/mask 128.91.39.0/25.
(ii) eth2 – 128.91.40.14 belongs to the subnet with IP/mask 128.91.40.0/25.

(iii) eth4 – 128.91.40.151 does not belong to any of the subnets specified (doesn't belong to 128.91.40.0/25 since the 25^{th} bit is 1); should be treated as default case.
(iv) eth3 – 191.2.153.18 belongs to the subnet with IP/mask 191.2.153.0/26.
(v) eth4 – 192.2.153.92 does not belong to any of the subnets specified (doesn't belong to 191.2.153.0/26 since the 26^{th} bit is 1); should be treated as default case.

(b) Consider that an organization has a CIDR IPv4 address pool 200.1.1.0/24. The network administrator wants to form four subnets for four departments, with humber of hosts as follows – (i) Subnet A – 75 hosts, (ii) Subnet B – 35 hosts, (iii) Subnet C – 20 hosts and (iv) Subnet D – 18 hosts. Give a possible arrangements to construct the subnets; provide the subnet IPs and the subnet masks for the four subnets. [5]

Answer: Subnet A requires 7 bits for addressing (since $2^6 < 75 < 2^7$), subnet B requires 6 bits for addressing (since $2^5 < 35 < 2^6$, subnet C and subnet D require 5 bits each for addressing (since $2^4 < 18 < 20 < 2^5$).

Now, such an allocation is impossible if all-zero and all-one subnets are not allowed. This is because atleast 2 extra bits are required to create the first subnet without all-zero and all-one allocations, i.e., either 01 or 10 is valid. This would leave us with 6 addressing bits (since we started with 8, which renders the allocation impossible.

If we assume that all-one subnets are allowed, the following allocation is possible:

IP Pool $\rightarrow$ 201.1.1.xxxx xxxx or 201.1.1.0/24
Subnet A $\rightarrow$ 201.1.1.1xxx xxxx or 201.1.1.128/25
Subnet B $\rightarrow$ 201.1.1.01xx xxxx or 201.1.1.64/26
Subnet C $\rightarrow$ 201.1.1.001x xxxx or 201.1.1.32/27
Subnet D $\rightarrow$ 201.1.1.011x xxxx or 201.1.1.96/27

Similar allocations are possible by replacing ones with zeros, if all-zero subnets are allowed.

4. (a) A router R receives a packet with source IP 191.160.2.2 and destination IP 202.141.81.6. The router has received the packet at its interface eth0 (with interface IP 128.96.171.92 and MAC 00:14:22:01:23:45). The router then makes a route lookup and finds out that the next hop is 128.98.10.1 and the interface to forward the packet is eth5 (with interface IP 128.98.10.20 and MAC 00:11:A2:F1:25:C2). The router then makes an ARP query and finds out the MAC for the IP 128.98.10.1 as 00:23:1F:7C:9D:A2. Write down the source IP, destination IP, source MAC and destination MAC for the outgoing packet from the router R. [4]

Answer:
Source IP: 191.160.2.2
Destination IP: 202.141.81.6
Source MAC: 00:11:A2:F1:25:C2 (eth5)
Destination MAC: 00:23:1F:7C:9D:A2

(b) What is the difference between routing information base and forwarding information base? [2]

Answer: The routing information base (RIB) is the routing table, as implemented in software, and is maintained in the control plane. It is dynamic and maintains entire routing information. In contrast, the forwarding information base (FIB) is like a cache for the RIB, and maintains a copy of only the required routes in the interface hardware (TCAM). The FIB is updated only when required (a route changes for the particular interface).

(c) Consider a MAC protocol where 10101011 is the SFD character and 11100111 is the ESC character. With this protocol, the sender wants to transmit following sequence of bits at the payload – 11001010 11100111 10101011 11100010 11100111 11100111 10110101. Write down the byte sequence that will go out from the sender after applying byte stuffing. [2]

Answer: The concept is that the ESC character is inserted before the SFD if SFD is part of the payload. Similarly, the ESC character is also inserted before itself, if ESC itself is part of the payload.

For the payload 11001010 11100111 10101011 11100010 11100111 11100111 10110101, SFD appears as part of the data in the 3^{rd} byte, whereas ESC appears as part of the data in the 2^{nd}, 5^{th}, and 6^{th} bytes. Therefore, the ESC character needs to be inserted before each of these characters.

The final payload will be: 11001010 ESC 11100111 ESC 10101011 11100010 ESC 11100111 ESC 11100111 10110101,
or 11001010 11100111 11100111 11100111 10101011 11100010 11100111 11100111 11100111 11100111 11100111 10110101.

(d) What is the requirement of spanning tree protocol at the MAC layer? Explain how it works. [3]

Answer: Sometimes, redundant links are deployed between switches to increase reliability and fault-tolerance. This creates loops in the topology. When switches flood frames for unknown destinations, the frames may keep looping in the LAN causing unnecessary overhead. A spanning tree protocol creates a loop-free logical topology, and avoids this problem.

In STP, each switch periodically broadcasts a configuration message out all of its ports, and processes the message it receives from other switches. Each switch has an identifier, based on its MAC address, and the one with the lowest identifier value is chosen as the root (using configuration message interactions). The switches then construct a tree of shortest paths from the root to every switch. Each switch remembers its shortest path to the root, and turns off ports that are not part of the spanning tree. The algorithm keeps running in order to take into account topology changes.

(e) What is the minimum Hamming distance for (i) detecting n bit errors, (ii) correcting n bit errors. Explain your answer. [2+2]

Answer: (i) Min. Hamming distance for n bit error detection is $n + 1$. This is because no set of n errors in a single bit could turn one valid codeword into some other valid codeword.

(ii) Min. Hamming distance to correct n bit error is $2n + 1$. This puts the valid codewords so far apart that even after bit errors in n of the bits, it is still less than half the distance to another valid codeword, so the receiver will be able to determine what the correct starting codeword was.