CS2105  **Tutorial 1** Answer paper

1. **[KR, Chapter 1, P6]** Consider two hosts, A and B, connected by a single link of rate ***R*** bps. Suppose that the two hosts are separated by ***m*** meters, and suppose the propagation speed along the link is ***s*** meters/sec. Host A is to send a packet of size ***L*** bits to Host B.
2. Express the propagation delay, ***dprop***, in terms of ***m*** and ***s***.

**dprop = m/s sec**

1. Determine the transmission time of the packet, ***dtrans***, in terms of ***L*** and ***R***.

**dtrans = L/R sec**

1. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay ***dend-to-end***.

**dend-to-end = dprop + dtrans sec**

1. Suppose Host A begins to transmit the packet at time t = 0. At time t = ***dtrans***, where is the last bit of the packet?

**The last bit just leaves Host A**

1. Suppose ***dprop*** is greater than ***dtrans***. At time t = ***dtrans***, where is the first bit of the packet?

**The first bit is flowing in the link and has not reached host B**

1. Suppose ***dprop*** is less than ***dtrans***. At time t = ***dtrans***, where is the first bit of the packet?

**The first bit has already reached host B**

1. Suppose ***s*** = 2.5 x 108, ***L*** = 120 bits, and ***R*** = 56 kbps. Find the distance ***m*** so that ***dprop*** equals ***dtrans***.

**m = (L/R) x s = 120 x 2.5 x 108 / (56 x 103) = 535714.2857 meters**

1. A packet switch receives a packet and determines the outbound link to which the packet should be forwarded. When the packet arrives, one other packet is halfway done being transmitted on this outbound link and four other packets are waiting to be transmitted. Packets are transmitted continuously in order of arrival. Suppose all packets are 1,500 bytes and the link rate is 2 Mbps.
2. What is the queuing delay for the packet?

**The arriving packet must first wait for the link to transmit 4.5\*1500 bytes = 6,750 bytes or 54,000 bits.** **Since bits are transmitted at 2 Mbps, the queuing delay is 27 msec.**

1. More generally, what is the queuing delay when all packets have length ***L*** (bits), the transmission rate is ***R***, ***x*** bits of the currently-being-transmitted packet have been transmitted, and ***n*** packets are already in the queue?

**Generally, the queuing delay is .**

1. **[Modified from KR, Chapter 1, P31]** In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message segmentation*. Figure 1.27 illustrates the end-to-end transport of a message with and without message segmentation.

Diagram

Description automatically generated

Consider a message that is 8 x 106 bits long that is to be sent from source to destination. Suppose each link in the figure is 2 Mbps. Ignore propagation, queuing, and processing delays.

1. Consider sending the message from source to destination ***without*** message segmentation. How long does it take to move the message from the source host to the first packet switch (router)?

**Time for source to send out the message = 8 x 106 / 2 x 106 = 4 sec. Since we assume no propagation delay, packet reaches the first switch at t = 4 sec.**

1. Following a), what is the total time to move the message from source host to destination host? Keeping in mind that each switch uses store-and-forward packet switching.

**The 1st switch needs to receive the entire message (at t = 4 sec) before it starts forwarding the packet onto the outgoing link. So does the 2nd switch.**

**With store-and-forward switching, the total time to move the message from source host to destination host = 4 + 4 + 4 = 12 sec.**

1. Now suppose that the message is segmented into 800 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

**Time to send the 1st packet to the 1st switch = 10000 / 2 x 106 = 5 msec. The source starts sending the 2nd packet at t = 5 msec. It takes another 5 msec to send this packet to the 1st switch. Time when the 2nd packet reaches the 1st switch is therefore 5 + 5 = 10 msec.**

1. How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part b) and comment.

**The 1st packet reaches destination at t = 15 msec. After that, every 5 milliseconds, one more packet arrives at the destination. Thus, time the 800th packet reaches the destination = 15 + 799 x 5 = 4010 msec = 4.01 sec.**

**It can be seen that end-to-end delay in using message segmentation (4.01 sec) is significantly less than sending a big file as one message (12 sec).**

1. In addition to reducing delay, what are reasons to use message segmentation?
2. **Without message segmentation, if bit errors are not tolerated and there is a single bit error, the whole message has to be retransmitted (rather than a single packet).**
3. **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.**
4. Discuss the drawbacks of message segmentation.
5. **Packets have to be put in sequence at the destination (network may re-order packets).**
6. **Message segmentation results in many smaller packets. Each packet needs to carry packet header of size tens of bytes (e.g. to specific destination address and port number). This is the header overhead of each packet to be discussed in later lectures.**
7. There are devices to be connected. There can be either 0 or 1 link between any 2 devices.
8. What is the minimum number of links needed to connect all devices?

**-1 links. Organize all devices into a tree topology, chain topology or star topology.**

1. What is the maximum number of links that can be used to connect all devices?

**x (-1) / 2 links. All devices connect to all other devices directly. This is known as mesh topology.**

1. What are the pros and cons of the network topologies in part a) and b)?

**a): Simple topology but failure of a single node or link partitions network. Also it tends to have longer paths between 2 nodes.**

**b): Most robust topology, 1 hop distance between all nodes, but is most expensive.**

CS2105  **Tutorial 2** Answer paper

1. Consider the following HTTP request message sent by a browser.

**GET** /index.html HTTP/1.1

**Host:** www.example.org

**Connection:** keep-alive

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36

**Accept-Encoding:** gzip, deflate

**…**

1. What is the URL of the document requested by this browser?

**www.example.org/index.html**

1. What version of HTTP is this browser running?

**HTTP version 1.1**

1. Does the browser request a non-persistent or a persistent connection?

**The browser requests a persistent connection, as indicated by the header field ‘Connection: keep-alive’.**

1. What is the IP address of the host on which the browser is running?

**[Tricky question] IP address is not shown in HTTP request message. One would be able to get such information from socket (to be learnt later).**

1. The text below shows the header of the response message sent from the server in reply to the HTTP GET message in Q1 above. Answer the following questions.

HTTP/1.1 200 OK

**Content-Encoding:** gzip

**Content-Type:** text/html; charset=UTF-8

**Date:** Wed, 23 Jan 2019 13:50:31 GMT

**Last-Modified:** Fri, 09 Aug 2013 23:54:35 GMT

**Connection**: Keep-Alive

**Content-Length:** 606

**…**

1. Was the server able to successfully find the document or not?

**The status code 200 and the phrase OK indicate that the server was able to locate the document successfully.**

1. What time did the server send the HTTP response message?

**The HTTP response message was formed on Wed, 23 Jan 2019 13:50:31 Greenwich Mean Time.**

1. How many bytes are there in the document being returned?

**There are 606 bytes in the document being returned.**

1. Did the server agree to a persistent connection?

**The server agreed to a persistent connection, as indicated by the header field ‘Connection: Keep-Alive field’.**

1. **[KR, Chapter 2, P1]** True or false?
2. A user requests a Web page that consists of some text and three images. For this page, the client will send one request message and receive four response messages.

**False. Download one object per request.**

1. Two distinct Web pages on the same Web server (for example, [www.mit.edu/research.html](http://www.mit.edu/research.html) and [www.mit.edu/students.html](http://www.mit.edu/students.html)) can be sent over the same persistent connection.

**True, because they are on the same server.**

1. The **Date:** header in the HTTP response message indicates when the object in the response was last modified.

**False. Header field ‘Date’ indicates the server response time. The time the object is last modified is denoted by another header field ‘Last-Modified’.**

1. HTTP response messages never have an empty message body.

**False. E.g. conditional GET whereby browser’s cached copy is up-to-date.**

1. **[Modified from KR, Chapter 2, P10]** Suppose within your Web browser, you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address.

Suppose that ***n*** DNS servers are visited before your host receives the IP address from DNS; visiting them incurs an RTT of ***DDNS*** per DNS server.

Further suppose that the Web page associated with the link contains ***m*** very small objects (in addition to the HTML page). Suppose the HTTP running is non-persistent and non-parallel. Let ***DWeb*** denote the RTT between the local host and the server of each object.

Assuming zero transmission time of each object, how much time elapses from when the client clicks on the link until the client receives all the objects?

**To map from hostname to IP address: n x DDNS (note: DNS is over UDP, so no need to establish connection).**

**To establish TCP connection and get the HTML page = DWeb + DWeb**

**To establish *m* TCP connection and get all *m* objects = m x (DWeb + DWeb)**

**Total time = n x DDNS + (m + 1) x 2 x DWeb**

1. **[Modified from KR, Chapter 2, P8]** Referring to the previous question, suppose that three DNS servers are visited. Further, the HTML file references five very small objects on the same server. Neglecting transmission delay, how much time elapses with:
2. Non-persistent HTTP with no parallel TCP connections?

**3 x DDNS + (5 + 1) x 2 x DWeb**

1. Non-persistent HTTP with the browser configured for five parallel connections?

**3 x DDNS + 2 x DWeb + 2 x DWeb**

**Need to fetch HTML file first (2 x DWeb). Subsequently the rest 5 objects can be fetched in parallel each using a TCP connection (2 x DWeb).**

1. Persistent HTTP with pipelining?

**3 x DDNS + 2 x DWeb + DWeb**

**Need to fetch HTML file first (2 x DWeb). The rest 5 objects can be fetched through the same TCP connection in parallel – no RTT for TCP handshake is needed.**

1. Do you know what is DNS cache poisoning? Search online for a real example.

**DNS cache poisoning (a kind of DNS spoofing) is a computer hacking attack, whereby rogue DNS records are introduced into a DNS resolver’s cache, causing the name server to return an incorrect IP address, diverting traffic to the attacker’s computer (or any other computer). For example, DDoS (Distributed Denial of Service Attack) on a particular machine can be achieved via DNS cache poisoning.**

**Examples:**

1. **DNS Poisoning in China:** [**http://www.howtogeek.com/161808/htg-explains-what-is-dns-cache-poisoning/**](http://www.howtogeek.com/161808/htg-explains-what-is-dns-cache-poisoning/)
2. **Angry Bird Website Defaced:** [**https://arstechnica.com/security/2014/01/angry-birds-website-defaced-following-reports-it-enables-government-spying/**](https://arstechnica.com/security/2014/01/angry-birds-website-defaced-following-reports-it-enables-government-spying/)

Now answer the following questions:

1. What is the status code returned from the server to your browser?

**Ans: 200**

1. When was the HTML ﬁle that you are retrieving last modiﬁed at the server?

**Ans: the value is denoted by the header ﬁeld ’Last-Modiﬁed’.**

CS2105  **Tutorial 3** Answer paper

1. Launch your browser and open its network diagnostic tool (e.g. press F12 if you use Chrome on Windows, or Cmd + Opt + I for Mac). Enter press the “Submit” button.
2. Look at the entry named “formResponse”. What is the HTTP request method issued?

**POST**

1. Briefly explain when HTTP POST and GET methods are used.

**(From Wikipedia) The POST request method requests that a web server accepts and stores the data enclosed in the body of the request message. It is often used when uploading a file or submitting a completed web form. In contrast, the HTTP GET request method is designed to retrieve information from the server.**

1. **[KR, Chapter 2, P21]** Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e. not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

**You may use ‘dig’ program to query the local DNS server. For example,**

**dig –t a www.abc.com**

**If IP address of this Web page has been queried by another computer seconds ago, your local DNS server should keep this knowledge in local DNS cache and is able to answer your query quickly. Otherwise, the query time will be long.**

1. **[Modified from KR, Chapter 2, P31]** You are given 4 programs: **TCPEchoServer.py**, **TCPEchoClient.py**, **UDPEchoServer.py** and **UDPEchoClient.py**.
2. Suppose you run **TCPEchoClient** before you run **TCPEchoServer**. What happens? Why?

**When creating a local socket, client attempts to make a TCP connection to a non-existent server process. Exception will be thrown.**

1. Suppose you run **UDPEchoClient** before you run **UDPEchoServer**. What happens? Why?

**UDP client doesn’t establish connection to server when creating local socket. Thus it works fine if you start client program first and then server program (but data sent to server are all lost).**

1. **[KR, Chapter 3, R7]** Suppose a process in Host C has a UDP socket with port number 6,789. Suppose both Host A and Host B each sends a UDP segment to Host C with destination port number 6,789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

**Yes, both segments will be directed to the same socket. Sock APIs are available for programmers to learn the IP address of the origin of a packet (e.g. in Java that is getAddress() method).**

1. **[Modified from KR, Chapter 3, P4]**
2. Suppose you have the following 2 bytes: **01011100** and **01100101**. What is the 1’s complement of the sum of these 2 bytes?

**Sum: 11000001, checksum: 00111110**

1. Suppose you have the following 2 bytes: **11011010** and **01100101**. What is the 1’s complement of the sum of these 2 bytes?

**Sum: 01000000, checksum: 10111111**

(Note: UDP and TCP use 16-bit words in computing their checksums. For simplicity you are asked to consider 8-bit checksums in this problem).

1. **[Modified from KR, Chapter 3, P5]** Suppose that UDP receiver computes the checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? You may use Q5 as an example to explain.

**If sender transmits the following two bytes: 01011100** **and** **01100101, and the two bits highlighted in red flip, then checksum remains unchanged and receiver will fail to detect this error.**

1. **[KR, Chapter 3, R9]** In our rdt protocols, why did we need to introduce sequence numbers?

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

CS2105  **Tutorial 4** Transport layer Answer paper

1. **[KR, Chapter 3, R6]** Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

**One would have to implement reliability checking and recovery mechanisms (ACK, seq #, checksum, timeout, re-transmission, etc.) at application layer. For example, sender needs to include relevant header/trailer fields in every packet (as illustrated below).**

|  |  |  |
| --- | --- | --- |
| IP Header | UDP Header | App Header fields + data + Trailer |

1. Show an example that if the communication channel between the sender and receiver can reorder messages (i.e. two messages are received in different order they are sent), then protocol **rdt3.0** will not work correctly.

A picture containing chart

Description automatically generated

1. **[KR, Chapter 3, P29]** It is generally a reasonable assumption, when sender and receiver are connected by a single wire, that packets cannot be reordered within the channel between the sender and receiver. However, when the “channel’ connecting the two is a network, packet reordering may occur. One manifestation of packet reordering is that old copies of a packet with a sequence or acknowledgement number of *x* can appear, even though neither sender’s nor receiver’s window contains *x*. With packet reordering, the channel can be thought of as essentially buffering packets and spontaneously emitting these packets at any point in the future. What is the approach taken in practice to guard against such duplicate packets?

**The approach taken in practice is to ensure that a sequence number is not reused until the sender is “sure” that any previously sent packets with the same sequence number are no longer in the network.**

**Firstly, TCP use large sequence number field (32-bit) to lower the chance a sequence number is to be reused. Secondly, a packet cannot “live” in the network forever. For example, IP protocol specifies TTL in packet header to ensure that datagrams do not circulate infinitely in the network. This field is decreased by one each time the datagram arrives at a router along the end-to-end path. If TTL field reaches 0, router will discard this datagram. In practice, a maximum packet lifetime of approximately three minutes is assumed in the TCP extensions for high-speed networks.**

1. **[Modified from KR, Chapter 3, P37]** Host A is sending data segments to Host B using a reliable transport protocol (either GBN or SR). Assume timeout values are sufficiently large such that all data segments and their corresponding ACKs can be received (if not lost in the channel) by Host B and the Host A respectively. Suppose Host A sends 5 data segments to Host B and the 2nd data segment is lost. Further suppose retransmission is always successful. In the end, all 5 data segments have been correctly received by Host B.

How many segments has Host A sent in total and how many ACKs has Host B sent in total if either GBN or SR protocol is used? What are their sequence numbers? Answer this question for both protocols.

**GBN: Host A sends 9 segment. They are initially sent segments 1, 2, 3, 4, 5 and later resent segments 2, 3, 4 and 5. Host B sends 8 ACKs. They are 4 ACKs with seq # 1 and 4 ACKs with seq # 2, 3, 4 and 5.**

**SR: Host A sends 6 segment. They are initially sent segments 1, 2, 3, 4, 5 and later resent segment 2. Host B sends 5 ACKs. They are 4 ACKs with seq # 1, 3, 4, 5 and 1 ACK with seq # 2 (for resent segment).**

1. **[KR, Chapter 3, R15]** Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 65; the second has sequence number 92.
2. How much data is in the first segment?

**92 – 65 = 27 bytes**

1. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

**65. Note that TCP acknowledgment is cumulative and states the expected in-order sequence number.**

1. **[KR, Chapter 3, P26]** Consider transferring an enormous file of *L* bytes from Host A to Host B. Assume an MSS of 512 bytes.
2. What is the maximum value of *L* such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field is 32 bits.

**TCP sequence number doesn’t increase by one with each TCP segment. Rather, it increases by the number of bytes of data sent. Therefore the size of the MSS is irrelevant here -- the maximum size file that can be sent from A to B without exhausting TCP sequence number is simply 232 bytes.**

1. For the *L* you obtain in (a), find how long it takes to transmit this file. Assume that a total of 64 bytes of transport, network, and data-link header are added to each packet before the resulting packet is sent out over a 155 Mbps link. Ignore flow control, congestion control and assume Host A can pump out all segments back to back and continuously.

**Number of packets = L / MSS = 232 / 512 = 8,388,608**

**64 bytes of headers will be added to each packet. Therefore,**

**Total bytes sent = 232 + 8388608 \* 64 = 4,831,838,208 bytes**

**Transmission delay = 4831838208 \* 8 / (155 \* 106) ≈ 249 seconds**

Answer the following questions:

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the ﬁle to gaia.cs.umass.edu?

**Ans: select an HTTP message sent from your computer to gaia.cs.umass.edu and explore the details (src-ip & src-port) of the TCP packet used to carry this HTTP message.**

1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

**Ans: ip: select an HTTP message explore the details of the TCP packet used to carry this HTTP message. port: 80.**

CS2105 **Tutorial 5** Network layer data plane Answer paper

1. **[KR, Chapter 4, R13]** What is the 32-bit binary equivalent of the IP address 202.3.14.25?

**11001010 00000011 00001110 00011001**

1. **[KR, Chapter 4, R25]** Suppose an application generates chunks of 40 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. Assume TCP header is 20 bytes and IP header is another 20 bytes, what percentage of each datagram will be overhead, and what percentage will be application data?

**IP datagram size is 80 bytes which consists of 40 bytes of header and 40 bytes of data. Thus percentage of overhead and data is each 50%.**

1. Combine the following three blocks of IP addresses into a single block:
2. 16.27.24.0/26
3. 16.27.24.64/26
4. 16.27.24.128/25

**16.27.24.0/24**

1. **[Modified from KR, Chapter 4, P16]**
2. Consider a subnet with network prefix 192.168.56.128/26. Give an example IP address (of form xxx.xxx.xxx.xxx) that belongs to this network.

**Any IP address in the range 192.168.56.128 to 192.168.56.191**

1. Suppose an ISP owns the block of addresses of the form 192.168.56.128/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the network prefixes (of form a.b.c.d/x) for the four subnets?

|  |  |
| --- | --- |
| **Network Prefix** | **Binary Expression** |
| **192.168.56.128/28** | **11000000 10101000 00111000 10000000** |
| **192.168.56.144/28** | **11000000 10101000 00111000 10010000** |
| **192.168.56.160/28** | **11000000 10101000 00111000 10100000** |
| **192.168.56.176/28** | **11000000 10101000 00111000 10110000** |

1. **[KR, Chapter 4, P7]** Consider a datagram network using 8-bit addresses. Suppose a router has the following forwarding table:

|  |  |
| --- | --- |
| **Prefix Match** | **Interface** |
| 11 | 0 |
| 101 | 1 |
| 100 | 2 |
| otherwise | 3 |

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

|  |  |  |  |
| --- | --- | --- | --- |
| **Prefix Match** | **Interface** | **IP Range** | **No. of IP** |
| 11 | 0 | **1100 0000 – 1111 1111** | **64** |
| 101 | 1 | **1010 0000 – 1011 1111** | **32** |
| 100 | 2 | **1000 0000 – 1001 1111** | **32** |
| otherwise | 3 | **0000 0000 – 0111 1111** | **128** |

1. What is private IP address? Does LumiNUS use private or public IP? When your laptop is connected to NUS network, does it receive a private or public IP?

**(Part of the following answer is extracted and modified from RCF1918:** [**https://tools.ietf.org/html/rfc1918**](https://tools.ietf.org/html/rfc1918)**)**

**The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of IP address space for private networks:**

**10.0.0.0 - 10.255.255.255 (10/8 prefix)**

**172.16.0.0 - 172.31.255.255 (172.16/12 prefix)**

**192.168.0.0 - 192.168.255.255 (192.168/16 prefix)**

**An enterprise that decides to use private IP addresses can do so without any coordination with IANA or an Internet registry. The address space can thus be used by many enterprises.**

**However, private IP addresses cannot have IP connectivity to any host outside of the enterprise. NAT or application layer gateways are needed to map private to public address and vice versa when traffic goes in and out private network.**

**Private IP is initially designed for experimentation purpose, but now used as a way to alleviate IPv4 address exhaustion. Its use is very common today.**

**LumiNUS use public IP address (137.132.10.10). Students may use ping command to check it. Laptops of students are assigned private IP addresses (e.g. 172.26.184.76).**

CS2105 **Tutorial 6** Network layer control plane Answer paper

1. **[Modified from KR, Chapter 4, P21]** Consider the network setup in the following figure. Suppose that the ISP assigns the router the address 24.34.112.235 and that the network address (i.e. network prefix) of this home network is 192.168.1/24.

A picture containing text, appliance

Description automatically generated

**192.168.1.5**

**192.168.1.9**

**192.168.1.2**

**192.168.1.1**

24.34.112.235

1. Give an example IP address assignment to all interfaces in this home network.
2. Suppose each host has two ongoing TCP connections, all to port 80 of a server at 128.119.40.86. Provide example corresponding entries in the NAT translation table.

|  |  |
| --- | --- |
| **NAT Translation Table** | |
| **WAN side** | **LAN side** |
| **24.34.112.235, 3000** | **192.168.1.5, 2105** |
| **24.34.112.235, 4000** | **192.168.1.5, 2106** |
| **24.34.112.235, 5000** | **192.168.1.2, 2105** |
| **24.34.112.235, 6000** | **192.168.1.2, 2108** |
| **24.34.112.235, 7000** | **192.168.1.9, 4000** |
| **24.34.112.235, 8000** | **192.168.1.9, 5000** |

1. **[Modified from KR, Chapter 4, P19]** Consider sending a 1500-byte IP datagram into a link that has an MTU of 500 bytes. Suppose the original datagram is stamped with the identification number 422. Also assume that IP header is 20 bytes long.
2. How many fragments will be generated?

**Data (segment) length = 1500 – 20 = 1480 (due to 20 bytes IP header)**

**Maximum size of data in each fragment = 500 – 20 = 480**

**Number of fragments = = 4**

1. What is the length of each fragment (including IP header)?

**1st – 3rd fragments have length 500. 4th fragment has length 60.**

1. What are the values of *identification number*, *offset* and *flag* in each fragment?

|  |  |  |  |
| --- | --- | --- | --- |
| **Fragments** | **ID Number** | **Offset** | **Flag** |
| **1** | **422** | **0** | **1** |
| **2** | **422** | **60** | **1** |
| **3** | **422** | **120** | **1** |
| **4** | **422** | **180** | **0** |

1. **[CS2105 Final Exam, April 2006]** The following diagram shows a simple network topology with 4 nodes. The links in the diagram are labeled with the cost of each link. The nodes run distance vector routing protocol. The protocol has terminated, and each node knows the cost of the minimum cost path to every other node.

z

20

w

y

3

x

3

7

6

2

The following table shows an incomplete distance vector table at node x. Fill in the missing distance vectors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **cost to w** | **cost to x** | **cost to y** | **cost to z** |
| **from x** | **3** | **0** | **3** | **5** |
| **from y** | **6** | **3** | **0** | **2** |
| **from z** | **8** | **5** | **2** | **0** |

Answer the following questions:

1. Within the IP packet header, what is the value in the upper layer protocol ﬁeld?

**Ans: ICMP.**

1. Which ﬁelds in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

**Ans: Identiﬁcation and Checksum.**

**Tutorial 7** Error detection / correction & collision detection Answer paper

1. **[KR, Chapter 5, R2]** If all the links in the Internet were to provide reliable delivery service, would the TCP reliable delivery service be redundant? Why or why not?

**IP datagrams in the same TCP connection can take different routes in the network, and therefore arrive at receiving host out of order. TCP is still needed to sort out received data in the correct order before passing them to application.**

**Also, IP datagrams can be lost due to routing loops, equipment failures, etc. For example, what if a router holding a frame crashes?**

1. **[KR, Chapter 5, P5, P6]** Consider a 4-bit generator with value **1001**, what is the CRC checksum if data has the following value?
2. **11000111010**

**110**

1. **01101010101**

**011**

1. **11111010101**

**011**

1. **10001100001**

**110**

1. Consider the following two-dimensional parity matrix of data.
2. Compute row sums, column sums and parity bit.
3. Give an example of a 1-bit error that can be detected and corrected.
4. Give an example of a 2-bits error that can be detected but cannot be corrected.
5. Give an example of a 4-bits error that cannot be detected.
6. There are many nodes in a shared medium network and most nodes are likely to transmit frequently. Which of the following multiple access protocol(s) is (are) suitable? (1) TDMA; (2) CSMA; (3) Token passing.

**TDMA and token passing are suitable because there is sufficient work to do to utilize the “fixed” resources allocated.**

**CSMA is not because many nodes competing for the shared channel can result in lots of collision. Utilization will be low.**

1. Nodes and are accessing a shared medium using CSMA/CD protocol, with propagation delay of 245 bit times between them (i.e., propagation delay equals to the amount of time to transmit 245 bits onto the link). Minimum frame size is 64 bytes. Suppose node begins transmitting a frame at bit time. Before finishes, node begins transmitting a frame. Assume no other nodes are active.

Write down your answers to the following 2 questions in the unit of **bit time**.

1. When is the latest time, by which can begin its transmission?

**The latest time B can begin transmission is before the signal from A reaches B, which is when bit time.**

1. Suppose begin its transmission at the time computed in a), can detects that has transmitted before it finishes transmission?

**Suppose begin transmission at bit time. Signal propagates to at bit time. is able to detect collision before it finishes transmission (at bit time).**

CS2105  **Tutorial 8** Link Layer Answer paper

1. **[KR, Chapter 5, R6]** In CSMA/CD, after the fifth collision, what is the probability that a node chooses ? The result corresponds to a delay of how many microseconds on a 10 Mbps Ethernet?

**After 5th collision, NIC will choose at random from {0, 1, 2, …, 25-1}. The chance to choose is 1/32.**

**NIC will wait for 4 \* 512 / 107 = 204.8 microseconds.**

1. **[Modified from KR, Chapter 5, P26]** Let’s consider the operation of a learning switch in the context of a network in which 4 nodes, labeled through , are star connected into an Ethernet switch (refer to the diagram on Lecture 9 notes page 37/38).

Suppose that the following events happened in sequence,

1. sends a frame to
2. replies with a frame to
3. sends a frame to

The switch table is initially empty. Show the state of the switch table after each of the above events (ignore TTL field). For each event, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers.

|  |  |  |
| --- | --- | --- |
| **Event** | **Switch table after event** | **Link(s) a frame is forwarded to** |
| sends a frame to | **(B, 4)** | **1, 2, 3** |
| replies with a frame to | **(B, 4) , (D, 3)** | **4** |
| sends a frame to | **(B, 4) , (D, 3)** | **1, 2, 4** |

1. Refer to the diagram on Lecture 9 notes page 15. Suppose nodes , and are star connected into a switch . , and are aware of the IP addresses of each other.
2. Consider sending an IP datagram from Host to Host . Suppose all of the ARP tables and switch table are up to date. Enumerate all the steps the host and switch take to move the packet from to .
3. **creates a frame with destination MAC address CC-49-DE-D0-AB-7D (’s address is found in ARP table).**
4. **This frame travels to switch and is forwarded towards (interface to is found in switch table).**
5. Repeat the problem in a), assuming that ARP table in the sending host is empty, but all other tables are up to date.
6. **broadcasts an ARP query packet, with destination MAC address FF-FF-FF-FF-FF-FF.**
7. **Switch forwards this ARP query packet to both and since destination MAC address is a broadcast address.**
8. **will ignore this ARP query packet but will reply to . Switch forwards the reply frame towards (interface to is found in switch table).**
9. **Subsequently can send IP datagram to as in part a).**
10. Repeat the problem in a), assuming that all tables in all nodes are empty.
11. **needs to issue an ARP query to know the MAC address of .**
12. **The query packet travels to switch and is forwarded to both and . Switch learns is reachable via the interface query packet arrives at.**
13. **will ignore this ARP query packet but will reply to . Switch forwards the reply frame towards (interface to is found in switch table). Switch learns is reachable via the interface reply frame arrives at.**
14. **Subsequently can send IP datagram to as in part a).**
15. Suppose sends an IP datagram to a host in another subnet. All of the ARP tables and switch table are up to date. Enumerate all the steps the host, switch and router take to move the packet to another subnet.
16. **creates a frame with destination MAC address E6-E9-00-17-BB-4B (’s address is found in ARP table).**
17. **This frame travels to switch and is forwarded towards (interface to is found in switch table).**
18. **checks the destination IP of the datagram and decides to forward it towards external network. It encapsulates the IP datagram in a new frame with source MAC address 1A-23-F9-CD-06-9B (dest MAC address not mentioned in question) and sends it through the interface towards external network.**

Answer the following questions:

1. Based on the contents of the Ethernet frame containing the HTTP GET message:
   1. What is the 48-bit Ethernet address of your computer?

**Ans: select HTTP GET message and explore the ’mac-src’ header ﬁeld in the ethernet frame used to carry this HTTP GET message.**

* 1. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? What device has this as its Ethernet address?

**Ans: destination address: select HTTP GET message and explore the ’mac-dst’ header ﬁeld in the ethernet frame used to carry this HTTP GET message.**

**No.**

**Gateway.**

1. Based on the contents of the Ethernet frame containing the ﬁrst byte of the HTTP response message:
   1. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu. What device has this as its Ethernet address?

**Ans: select HTTP response message and explore the ’mac-src’ header ﬁeld in the ethernet frame used to carry this HTTP response message.**

**Neither my computer nor gaia.cs.umass.edu.**

**Gateway.**

* 1. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

**Ans: select HTTP response message and explore the ’mac-dst’ header ﬁeld in the ethernet frame used to carry this HTTP response message. Yes.**

CS2105  **Tutorial 9** Media Streaming Answer paper

1. **[KR, Chapter 9, R2]** There are two types of redundancy in video. Describe them, and discuss how they can be exploited for efficient compression.

**Spatial redundancy: Redundancy within the same image, e.g. consecutive pixels with the same color. We could compress the video frame by sending just the color value and its count (instead of the color value *N* times).**

**Temporal redundancy: Redundancy between multiple images, e.g. consecutive frames that are very similar and only have a small change between them. We could compress the second video frame by sending only its difference from the original frame (instead of the full uncompressed frame).**

1. **[KR, Chapter 9, R3]** Suppose an analog audio signal is sampled 16,000 times per second, and each sample is quantized into one of 1,024 levels. What would be the resulting bit rate of the PCM digital audio signal?

**Each sample would require bits encode its data. Hence, sampling at 16,000 times per second would generate a signal of 160,000 bits per second = 160 kbps.**

1. **[KR, Chapter 9, R7]** With HTTP streaming, are the TCP receive buffer and the client’s application buffer the same thing? If not, how do they interact?

**No, they are not the same. TCP and application are at different layers of the OSI model. TCP receive buffer (at transport layer) receives the TCP packets from the network layer, which are then processed and forwarded as HTTP packets to the client’s application buffer (at the application layer). The client (e.g. Chrome browser), upon receiving these HTTP packets, then processes and renders the video playback to the user.**

1. **[KR, Chapter 9, R10]** Why is a packet that is received after its scheduled playout time considered lost?

**When an application receives a packet that is *late*, i.e. received after its scheduled playout time, the application would not be able to play that packet anymore. Hence, from the perspective of the application, the packet has been lost.**

1. **[Modified from KR, Chapter 9, R12]** How are different RTP streams from within the same session identified by the receiver? How are RTP and RTCP packets differentiated?

**RTP streams in the same session are identified using the SSRC field.**

**RTP and RTCP packets use distinct port numbers.**

1. In practice, RTP tends to be used over UDP while RTSP tends to be used over TCP. Why might this be so?

**RTP sends media flow (e.g. multiple video packets) while RTSP sends playback commands (e.g. “Play” and “Pause”). Hence, it may be preferred that RTP prioritizes speed over reliability (UDP), so that the client could playback the video with minimal delay (losing a few video packets in-between is a relatively small cost since its impact on playback experience is generally tolerable). On the other hand, it may be preferred that RTSP prioritizes reliability (TCP), so that we do not lose any of these commands requested by the client.**

1. **[KR, Chapter 9, P11]** Consider the figure below (which is similar to Lecture 10 notes page 26). A sender begins sending packetized audio periodically at . The first packet arrives at the receiver at .

A close up of a piece of paper

Description automatically generated

* 1. What are the delays (from sender to receiver, ignoring any playout delays) of packets 2 through 8? Note that each vertical and horizontal line segment in the figure has a length of 1, 2, or 3 time units.

**Packet 2: 7 units, Packet 3: 9 units, Packet 4: 8 units,**

**Packet 5: 7 units, Packet 6: 9 units, Packet 7: 8 units, Packet 8: 8 units**

* 1. If audio playout begins as soon as the first packet arrives at the receiver at , which of the first eight packets sent will *not* arrive in time for playout?

**Packet 3, 4, 6, 7, 8**

* 1. If audio playout begins at , which of the first eight packets sent will not arrive in time for playout?

**Packets 3, 6**

* 1. What is the minimum playout delay at the receiver that results in all of the first eight packets arriving in time for their playout?

**Minimum playout delay is**

1. **[Modified from KR, Chapter 9, P13]** Recall the two FEC schemes for VoIP described in lecture. Suppose the first scheme (Scheme 1) generates a redundant chunk for every four original chunks. Suppose the second scheme (Scheme 2) uses a low-bit rate encoding whose transmission rate is 25 percent of the transmission rate of the nominal stream. (Note: we ignore the effects of playout delay in this question as we assume that all packets, including FEC packets, will be received prior to reconstruction and playback)
   1. How much additional bandwidth does each scheme require?

**Scheme 1: Every 4 original chunks will have 1 redundant chunk = 25% additional bandwidth; Scheme 2: Every chunk will have its redundant low-quality chunk “piggyback” on the next transmission = 25% additional bandwidth. Hence, both schemes will require additional 25% bandwidth.**

* 1. How do the two schemes perform if the first packet is lost in every group of five packets? Which scheme will have better audio quality?

**Scheme 1 will be able to reconstruct the original high-quality audio, while Scheme 2 will get a low-quality audio packet in every 5 packets. Hence, Scheme 1 will have better overall audio quality.**

* 1. How do the two schemes perform if the first packet is lost in every group of two packets? Which scheme will have better audio quality?

**Scheme 2 will have better audio quality as Scheme 1 is unable to recover from the packet losses while Scheme 2 can (albeit with low-quality). Note, redundancy-based FEC (e.g., XOR in Scheme 1) has a fixed loss limit up to which we can recover the data. If the losses exceed that limit, then all data lost in that group cannot be recovered.**

CS2105  **Tutorial 10** Security Answer paper

1. [KR, Chapter 8, P1] Using the substitution cipher (monoalphabetic cipher) shown in Lecture 11, on page 14, notes:
   1. encode the message “this is a secret message”

**uasi si m icbocu hciimzc**

* 1. decode the message “tcow ihmou”

**very smart**

1. [KR, Chapter 8, R6] Suppose 𝑁 people each want to communicate with 𝑁−1 other people. All communication between any two people, 𝑖 and 𝑗, is visible to all other people but no other person should be able to decode their communication. In total, how many keys are required in this group if:
   1. Symmetric key encryption is used in each communication?

**There are pairs of people and each pair needs to share a symmetric key. The total number of keys is .**

* 1. Public key encryption is used in each communication?

**With public key encryption, each person has a public key which is known to all, and a private key which is secret and known to the user only. There are thus**  **keys.**

1. [KR, Chapter 8, P13] In the BitTorrent P2P file distribution protocol, the seed breaks a file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a small subset of peers in the torrent. These unsuspecting peers then redistribute the bogus blocks to other peers, which in turn redistribute the bogus blocks to even more peers. Thus, it is critical for BitTorrent to have a mechanism that allows a peer to verify the integrity of a block, so that it doesn’t redistribute bogus blocks.  
   Assume that when a peer joins a torrent, it initially gets a .torrent file from a *fully trusted* source. Describe a simple scheme that allows peers to verify the integrity of blocks.

**A file is broken into a number of blocks of identical size. For each block, a hash is calculated (e.g., using MD5 or SHA-1). The hashes for all of the blocks are saved in the .torrent file.**

**When a block is downloaded, a peer calculates the hash of this block and compares it to the recorded hash in the .torrent file. If the two hashes are equal, this block is error-free. Otherwise, the block is bogus and should be discarded.**

1. Suppose Alice wants to send a secure email 𝑚 to Bob, and wants to ensure its confidentiality and integrity. Alice performs the following steps (Figure 8.21 on textbook which is reproduced below):

![A picture containing diagram

Description automatically generated]()

1. generates a random session key
2. encrypts the session key with Bob’s public key , obtaining (KS)
3. hashes the message 𝑚 with a cryptographic hash function 𝐻, obtaining message digest 𝐻(𝑚)
4. encrypts the hash with Alice’s private key , obtaining digital signature (𝐻(𝑚))
5. encrypts the message 𝑚, concatenated (⊕) with (𝐻(𝑚)), using the session key to obtain ( 𝑚⊕(𝐻(𝑚)) )
6. finally, sends ( 𝑚⊕(𝐻(𝑚)) )⊕ () to Bob

Show what Bob has to do to verify that is indeed crafted by Alice and has not been modified during transmission.

**Bob has to:**

1. **compute**  **to recover the session key using Bob’s private key**
2. **with , Bob decrypts the message , gets** 𝒎 **and**
3. **use Alice’s public key**  **to recover :**
4. **with** 𝒎**, Bob computes**  **and verifies that it is equal to from step 3**

**NOTE: symmetric key crypto () is used to encrypt** 𝒎**, instead of public key crypto because public key crypto is much slower.**

**Since only Bob has his private key, the intruder cannot decrypt**  **and therefore cannot get the session key . Without , the intruder cannot decrypt** 𝒎**.**

**The intruder cannot impersonate Alice, since Alice digitally signed the message with her private key .** **Otherwise, Bob cannot get the right**  **when trying to decrypt with Alice's public key.**

**The intruder cannot tamper with** 𝒎**, since modifying** 𝒎 **would cause its hash to be different and Bob would detect this.**