Introduction

Network Transport Times

1 . . . . *10 points*

(a) (2 points) traceroute and ping are useful command line tools to show the network path and latency to another host.

Using ping, find the average round trip time between Iceland and:

1. New York (96.110.41.133)
2. Tokyo (180.87.181.93)

Provide a screen shot of your ping results.

(b) (4 points) Solve the following problem using the times you found above. A server in Iceland is being used for a game of online bullet chess by two players, one in New York and one in Tokyo. In bullet chess, each player has a limited time to make all their moves, in this case 1 minute each in total for all their moves. Apart from the usual chess rules, a player looses when they run out of time to make their move. Assuming both players take the same amount of time to make their moves locally, and neither is able to win or draw the game within their time limit, which player will win, and by how many seconds?

(c) (2 points) Using the ping command, what is the round trip time (RTT) to the following hosts?

1. mel1.speedtest.telstra.net
2. per1.speedtest.telstra.net

(d) (2 points) Both the hosts are in Australia, mel1 is in Melbourne, perl1 is in Perth, and are usally connected by a fibre-optic cable. (We have occasionally seen this link fall back on satellite in the past, in that case calculate the approximate orbital height of the satellite.) If the speed of light in a vacuum is 300,000,000 m/s and the core index of refraction of fiber-optic cable in the Australian backbone is 1.50, approximately how far is Perth from Melbourne?

1. New York (96.110.41.133)

```
[onni2@onniArch ~]$ ping -3 96.110.41.133
PING 96.110.41.133 (96.110.41.133) 56(84) bytes of data.
64 bytes from 96.110.41.133: icmp_seq=1 ttl=241 time=121.110 ms
64 bytes from 96.110.41.133: icmp_seq=2 ttl=241 time=169.300 ms
64 bytes from 96.110.41.133: icmp_seq=3 ttl=241 time=264.117 ms
64 bytes from 96.110.41.133: icmp_seq=4 ttl=241 time=169.395 ms
64 bytes from 96.110.41.133: icmp_seq=5 ttl=241 time=192.450 ms
64 bytes from 96.110.41.133: icmp_seq=6 ttl=241 time=214.300 ms
64 bytes from 96.110.41.133: icmp_seq=7 ttl=241 time=128.650 ms
64 bytes from 96.110.41.133: icmp_seq=8 ttl=241 time=128.321 ms
64 bytes from 96.110.41.133: icmp_seq=9 ttl=241 time=179.600 ms
64 bytes from 96.110.41.133: icmp_seq=10 ttl=241 time=203.927 ms
64 bytes from 96.110.41.133: icmp_seq=11 ttl=241 time=118.899 ms
^C
--- 96.110.41.133 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10012ms
rtt min/avg/max/mdev = 118.899/171.824/264.117/43.672 ms
[onni2@onniArch ~]$
```

2. Tokyo (180.87.181.93)

```
[onni2@onniArch ~]$ ping -3 180.87.181.93
PING 180.87.181.93 (180.87.181.93) 56(84) bytes of data.
^C
--- 180.87.181.93 ping statistics ---
13 packets transmitted, 0 received, 100% packet loss, time 12268ms
```

It did not want to ping that address so i found another in Tokyo

```
[onni2@onniArch ~]$ ping -3 137.220.233.208
PING 137.220.233.208 (137.220.233.208) 56(84) bytes of data.
64 bytes from 137.220.233.208: icmp_seq=1 ttl=104 time=292.817 ms
64 bytes from 137.220.233.208: icmp_seq=2 ttl=104 time=547.229 ms
64 bytes from 137.220.233.208: icmp_seq=3 ttl=104 time=321.929 ms
64 bytes from 137.220.233.208: icmp_seq=4 ttl=104 time=351.808 ms
64 bytes from 137.220.233.208: icmp_seq=5 ttl=104 time=374.047 ms
64 bytes from 137.220.233.208: icmp_seq=6 ttl=104 time=390.130 ms
64 bytes from 137.220.233.208: icmp_seq=7 ttl=104 time=279.537 ms
64 bytes from 137.220.233.208: icmp_seq=8 ttl=104 time=327.786 ms
64 bytes from 137.220.233.208: icmp_seq=9 ttl=104 time=357.011 ms
^C
--- 137.220.233.208 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8011ms
rtt min/avg/max/mdev = 279.537/360.254/547.229/74.319 ms
```

b) $60s = 1000\ ms$

T = 171,824 + 1000 = 1171,824ms | 60,000 ÷ 1171,824 = 51,2 moves 349.2

N = 360,254 + 1000 = 1360,254 ms | 60.000 ÷ 1360,254 = 44,1 moves 166.5

60s = 60000ms

then lets assume one move takes 1000ms. then if we add the rtt to 1000 we get 1171.824ms and 1360.254ms then devide that with 60000 to see how many moves they would be able to do

New york could do 51.2 moves and Tokyo could do 44.1 moves. SO 44.1 times 1171.824 is 51.641s when tokyo hits it limit so new york wins with 8.358s left if 1000 is the time if 0 then he would win with 31.391304s left

44,1 x 1171 = 51,641 Seconds when Tokyo hits its limit.

for 0s per move it would be 28.6s

So New york wins with 60.000 − 51,641 = 8,358 seconds left

```
--- mel1.speedtest.telstra.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6227ms
rtt min/avg/max/mdev = 351.342/412.638/522.259/50.742 ms
```

```
--- per1.speedtest.telstra.net ping statistics ---
11 packets transmitted, 10 received, 9.09091% packet loss, time 10024ms
rtt min/avg/max/mdev = 387.459/440.204/569.458/51.013 ms
```

$\frac{3 \cdot 10^8}{1,5} = 2 \cdot 10^8$ $\qquad \frac{412}{2} = 207$ $\qquad \frac{440}{2} = 220$

$2 \cdot 10^8 \cdot (220 - 207) = 2 \cdot 10^8 (0{,}013) = 2600 km$

i get 13ms but that in but i need it in s

$2*10^8*(220-207) = 2*10^8(0.013) = 2600km$

Network Throughput

2 . . . . *7 points*

(a) (2 points) You need to transfer a geophysical dataset of 100TB stored on disk in Iceland to the Norwegian Metrology Office. How long will it take to transfer this dataset to Norway assuming a 1Gbps connection, and 15% protocol overhead?

(b) (2 points) Ref: `https://en.wikipedia.org/wiki/Linear_Tape-Open`

An industry standard tape (circa 2018) can hold 12TB of data on a single cartridge. Assuming a best case scenario of 3 hours ground transport time to Keflavik airport and 1 hour from Oslo to destination company, with a scheduled flight time also of 3 hours. How much data do you need before it is quicker to send the data by tape than transfer it over the network? (Ignore time to read and write the tape.)

(c) (2 points) Tannenbaum in Computer Networks wisely advises never to overlook the speed of sending data by existing transport networks - planes in this case. However, his example overlooks the time taken to create the tapes in the first place.

Assuming that the maximum writing and reading speed for a tape is 900(MB/s), how long does it actually take to transfer the data to Norway including the time to read and write the tapes, and that you only have one tape reader/writer in each city?

(d) (1 point) What is the new break even amount for sending data by planes?

Network Engineering

3 . . . . *6 points*

An ISP is statistically multiplexing its customers over 50Gbs links. You have been asked to calculate how many customers it can afford to assign to each link, and maintain a reasonable level of service to each one.

Nominally, each customer is being sold a 1Gbps link.

(a) (1 point) If each customer is to be guaranteed access to 1Gbps at any time, how many customers can the ISP provision per 50Gbps link.

(b) (2 points) Assume that the 50Gbps link costs the ISP 5,000,000 ISK/month, and the ISP needs to make 25% profit to cover all overheads. What is the smallest number of customers that the ISP can provision for each 1Gbps link, and still meets its profit targets, if the ISP charges each customer 10,000ISK for their Internet service?

(c) (1 point) What is the maximum speed each customer will be able to download data at, assuming all the customers provisioned in (b) are maximising their network connection?

(d) (2 points) The ISP decides that on average each business customer will use their link 20% of the time, evenly distributed over the day, and that households will only

2a) 100TB → 100000Gb

1Gbps -> 0.850Gbps/8 = 106.25MB/s =0.10625GB/s
100TB/0.00010625TB/s=941176.4076/3600=261.44kl / 24 = 10.9 dagar

2b)
$3+1+3=(7\cdot 60)\cdot 60$
$\text{Max data} = \text{Time(s)} \cdot (\text{GB/s}) = 25200 \cdot \left(\frac{0{,}85}{8}\right) = \boxed{2677{,}5\ \text{GB}}$

2677.5GB

2,7TB

12000000

2c)
$12T \rightarrow \frac{12000000\ \text{MB}}{900\ \text{MB/s}} = \frac{13333\,s}{60} = 3{,}7h$
$7h + (3{,}7 \cdot 2) = \boxed{14{,}4h}$

14,4 hours

2d)
$14{,}4h \rightarrow 51840\,s \cdot \left(\frac{0{,}85}{8}\right) = 5508\ \text{GB}$

5508GB

3a)
50/1 = 50 so he can have 50 customers and still have 1Gbps
3b)
they need to make 6250000 for them to have profit aka 25% more than 5000000.
charging 10000 each customer for 1Gbps.
but that means for them to reach the profit they need 6250000/10000 =625 customers
that is a lot more than 50
625/50 = 12.5

3c)
1Gbps/13 = 0.0769Gbps aka 80Mbps for every customer if they are maximizing
3d)
then we know that businesses will use 20% of 1Gbps aka 0,2Gbps. could that mean that 50/0.2 which is 250 people is the answer?

use the Internet in the evenings, and place less load on their connections. Assuming this is correct, how many customers can the ISP now provision and still maintain the illusion that they have access to 1Gbps each?

Error Detection / Correction

4 . . . . *7 points*

(a) (1 point) Shortly explain the purpose of redundancy such as parity bits or Hamming codes in data transmission.

(b) (2 points) Given the following sequence of data bits (in groups of 7), determine the even parity bit that should be added for error detection for each group.

data	parity bit
1011001	
1100001	
0000000	
1000001	

(c) (2 points) For a transmissing you are using the Hamming(7,4) code from slides 33-36 in lecture 4. You receive the word 1110011 (4 data bits and 3 parity bits). Identify if there is an error and locate the erroneous bit! What was the orginal data that was sent?

(d) (2 points) Can you be absolutely sure about the answer to the previous question? Explain!

Protocol Overhead

5 . . . . *9 points*

Your company has introduced a new high-speed low-latency intra-office messaging system. Internally, the system sends messages using UDP to the recipients desktop computer, all of which are connected via Ethernet to the same local network. You fancy a colleague on the same office floor and send that colleague the message ”Hi, how are you today?”.

(a) (2 points) Including protocol overhead, how many bytes are sent over Ethernet to deliver this message?

(b) (1 point) What is the percentage of overhead in the total communication?

(c) (2 points) Suppose you wanted to minimize overhead in the communication, what is the optimal length of message to reduce the relative overhead?

(d) (1 point) What is the percentage of overhead in that case?

(e) (2 points) Suppose this is the first communication between your computer and the recipients computer since turning your computer on this morning. What other packets need to be sent and received by your computer before it can send the actual message?

(f) (1 point) How much overhead do these extra packets add?

4. It is for error detection and error correction. Parity bits are calculated and appended to original data before transmission. They do not carry any data persei. They are like a error detection, so on the recieving end they are do the same calculation and if it is not correct an error has occurred.
Hamming goes one step further. it does the same but with many redundant bits so you can pinpoint where exactly the corrupted bit is!

b)

data	parity bit
1011001	10110010
1100001	11000011
0000000	00000000
1000001	10000010

c)

1110011

p1 1,3,5,7 =1
p2 2,3,6,7 =0
p3 4,5,6,7 = 0

compare errors
p3p2p1
001 = first bit!

So correct code would be
0110011

d)
i think i explained it quite well in c.
i checked wich parity bit was even and which one was odd. odd ones are incorrect.
I am double checking and i think the point of question d is he does not specify what system is sending.
so the but too flip could be 100 since some systems have it the other way so teh bit to flip is 4 also
no becouse if there are mulpiple bits bad/wrong then it might be wrong

5a)

protocol	size
UDP	8 bytes
ipv4	minum 20 bytes up to 60 bytes
ethernet H	14 bytes
ethernet T	4 bytes
total	46 bytes
string	22 bytes

String = "Hi, how are you today?"
the string bytes is 22 unless we are also counting "\0" then it is 23 bytes. but for this i will be using 22 bytes since it does not say the null terminated is in it and it does not say the programming language we are using

answer would be 46 + 22 = 68 bytes are sent

5b)
the total pecentage of the overhead is 69% and the string itself is only 31%
5c)
The largest MTU for ethernet is 1500. So i am guessing we are using that as a base.
So the largest messaeg to send without fragmentation and keep the overhead is 1500 bytes - 46
1454
5d) which will make the overhead 3.2% of the full message
5e and f)
i am assuming you mean ARP send and reply
since it is again going over ethernet using ipv4
the overhead for a send and reply for arp is 46.
ethernet header 14 bytes
arp payload 28 bytes
and ethernet tailer 4 bytes
equals 46 bytes.
that times 2 is 92 bytes

ARP and92 bytes

I did think that the payload would inly be 42 since that is what wireshark showed me but then i read that wireshark does not count for the tailer.