CS408 – Computer Networks – Spring 2022

Homework 2 –Network problems from the Application and Transport layers and general ones

Oytun Kuday Duran 28357

1) (10 points) In this problem, we consider sending real-time voice from Host A to Host B over a packet-switched network (VoIP). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 56-byte packets and when a packet is formed, it is send to host B. There is one link between Hosts A and B; its transmission rate is 2 Mbps and its propagation delay is 10 msec. What's the elapsed time since when the first bit is formed (from the original analog signal at Host A) until the bit is received at host B?

ANSWER:

56 byte packets over 64 kbps bit stream =>

56 byte = 56 * 8 = 448 bits

64 kbps (kilobit per second) = 64 * 10^3 = 64000 bit per second.

448/64000 = 0.007 seconds it takes conversion from analog to digital bit stream.

Transfer rate is 2 mbps which is 2*10^6 =2 000 000 bits per second that it transfers.

We have 56 * 8 = 448 bits to transfer.

448/2*10^6 = 0.000224 seconds it takes to transfer packets between hosts.

Propagation delay is 10 ms which is 0.01 seconds.

The total time it takes is = Propagation delay + transfer time + conversion time =

= 0.007 + 0.000224 + 0.01 = 0.017224 seconds

2) (32 points) Suppose two hosts, A and B, are separated by 20,000 kilometers and are connected by a direct link of R = 2 Mbps. Suppose the propagation speed over the link is 2.5 · 10

8 m/sec.

a) Calculate the propagation (delay) time $tprop$

b) Calculate the bandwidth-delay product, i. e. simply do $R \cdot tprop$

c) What is the maximum number of bits that will be in the link (propagating from Host A to Host B) at any given time?

d) How long does a single bit propagate in meters for the time needed to transmit a single bit (known as the width - in meters - of a bit in the link)?

e) Derive a general expression for the width of a bit in terms of the propagation speed s, the transmission rate R, and the length of the link m. You can use d) for some help and hints.

f) A file of 100 KB (1KB = 1000 B) is send from host A to host B. After Host B receives the whole file, it sends a single acknowledgment (ACK) to Host A. How long does it take for Host A to receive the ACK from Host B after sending the file, assuming that the file is send continuously? ACK size is neglectable.

g) Suppose now the file of 100 KB is broken up into 20 packets with each packet containing 40,000 bits. Suppose that each packet is acknowledged by the receiver and the transmission time of an acknowledgment packet is negligible. Finally, assume that the sender cannot send a packet until the preceding one is acknowledged (stop and wait). How long does it take to send the file? You can neglect header sizes.

h) compare the results from f) and g)

ANSWER:

a) tprop = distance /propagation speed = (km to meters->) $20000*10^3/2.5 * 10^8$= 0.08 seconds

b) R * tprop = $2*10^6 *0.08 =1.6 * 10^5$ bits.

c) The bandwidth delay product is $1.6 * 10^5$ bits which equals to 160000 bits.

d) dist/BW delay product = $20000*10^3 / 160000 = 5/4 * 10^3 = 1.25 * 10^3$ = 1250 meters.

e) width of a bit = dist / BW delay product = dist / (R*tprop) =dist / (R*dist/ propagation speed) = propagation speed / R where R is transmission rate so s/R

f) 100 KB is 100000 B. The time it takes + 0.08 sec prop + 0.08 ack prop delay so $100000*8/2*10^6$= 0.4 sec to transfer.

0.4 +0.08 + 0.08 = 0.56 seconds.

g) 20*2*0.08 (waiting for each packages prop and ack prop.) + 0.5 since overall transfer time doesn't change since overall file size doesn't change so 3.2 + 0.4 = 3.6 seconds.

h) Dividing a file to many packages seems like increases the time it takes to send files a lot because of propagation delay and acknowledgements per packages.

3) (8 points) 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; the successive visits incur an RTT of $RTT1$, $RTT2$, … , $RTTn$. Further suppose that the Web page (consisting of a small amount of HTML text) associated with the link contains exactly eight small objects of neglectable size. Let $RTT0$ denote the RTT between the local host and the server containing the Web page and the objects. Assuming zero transmission time per object, how much time elapses from when the client clicks on the link until the client receives the object if we use:

a) Non-persistent HTTP with no parallel TCP connections?

b) Non-persistent HTTP with the browser configured for 5 parallel connections?

c) Persistent HTTP with no parallel connections?

d) Persistent HTTP with 8 parallel connections?

ANSWER:

a) 2 * $RTT0$ + 2*8 * $RTT0$ + $RTT1$ + $RTT2$ + … + $RTTn$ = 18 * $RTT0$ + $RTT1$ + $RTT2$ +…+ $RTTn$

(Setting up connection and querying the dns first, then requesting 8 files 1 by 1)

b) 2 * $RTT0$ + 2*2 $RTT0$ + $RTT1$ + $RTT2$ + …+ $RTTn$ = 6 * $RTT0$ + $RTT1$ + $RTT2$ + …+$RTTn$

(Setting up connection and querying the dns first, *8 small objects and 5 each time for 2 times -3 object at second request)*

c) 2 * $RTT0$ + $RTT0$ + $RTT1$ + $RTT2$ + … + $RTTn$ = 3 * $RTT0$ + $RTT1$ + $RTT2$ +…+ $RTTn$

(Setting up connection and querying the dns first, request/Receive *8 small objects at once with 1 request)*

Important: I couldn't be sure whether the packages sent at once since I couldn't find a similar example from lectures. It also could be 2 * $RTT0$ + 8* $RTT0$ + $RTT1$ + $RTT2$ + … + $RTTn$ = 10 * $RTT0$ + $RTT1$ + $RTT2$ +…+ $RTTn$ *for 8 packages.*

d) 2 * $RTT0$ + $RTT0$ + $RTT1$ + $RTT2$ + … + $RTTn$ = 3* $RTT0$ + $RTT1$ + $RTT2$ +…+ $RTTn$

(We establish once. Then 1 time => 8 packages at once)

4) (20 points) Consider distributing a file of F= 15 $Gbits$ to N peers. The server has an upload rate of us =30 Mbps, and each peer has a download rate of di = 2 Mbps and an upload rate of u. For N=10, 100, and 1,000 and u = 300 Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution. Show your work!

ANSWER:

Using formulas:

D_{cs} = max{N*F/u_s , F/d_{min}}

D_{P2P} = max{F/u_s , F/d_{min}, N*F/(u_s + Σu_i)}

For n=10 and u = 300 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 300/1000 = 0.3 Mbps

D_{cs} = max{10*15000/30 , 15000/2} = max{5000,7500} = 7500 sec

D_{P2P} = max{15000/30 , 15000/2, 10*15000/(30 + 10*0.3)} = max{500,7500,4545} = 7500 sec

For n=100 and u = 300 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 300/1000 = 0.3 Mbps

D_{cs} = max{100*15000/30 , 15000/2} = max{50000,7500} = 50000 sec

D_{P2P} = max{15000/30 , 15000/2, 100*15000/(30 + 100*0.3)} = max{500,7500,25000} = 25000 sec

For n=1000 and u = 300 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 300/1000 = 0.3 Mbps

D_{cs} = max{1000*15000/30 , 15000/2} = max{500000,7500} = 500000 sec

D_{P2P} = max{15000/30 , 15000/2, 1000*15000/(30 + 1000*0.3)} = max{500,7500,45454} = 45454 sec

For n=10 and u =700 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 700/1000 = 0.7 Mbps

D_{cs} = max{10*15000/30 , 15000/2} = max{5000,7500} = 7500 sec

D_{P2P} = max{15000/30 , 15000/2, 10*15000/(30 + 10*0.7)} = max{500,7500,4054} = 7500 sec

For n=100 and u = 700 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 700/1000 = 0.7 Mbps

D_{cs} = max{100*15000/30 , 15000/2} = max{50000,7500} = 50000 sec

D_{P2P} = max{15000/30 , 15000/2, 100*15000/(30 + 100*0.7)} = max{500,7500,15000} = 15000 sec

For n=1000 and u = 700 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 700/1000 = 0.7 Mbps

D_{cs} = max{1000*15000/30 , 15000/2} = max{500000,7500} = 500000 sec

D_{P2P} = max{15000/30 , 15000/2, 1000*15000/(30 + 1000*0.7)} = max{500,7500,20548} = 20548 sec

For n=10 and u = 2000 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 2000/1000 = 2 Mbps

D_{cs} = max{10*15000/30 , 15000/2} = max{5000,7500} = 7500 sec

D_{P2P} = max{15000/30 , 15000/2, 10*15000/(30 + 10*2)} = max{500,7500,3000} = 7500 sec

For n=100 and u = 2000 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 2000/1000 = 2 Mbps

D_{cs} = max{100*15000/30 , 15000/2} = max{50000,7500} = 50000 sec

D_{P2P} = max{15000/30 , 15000/2, 100*15000/(30 + 100*2)} = max{500,7500,6521} = 7500 sec

For n=1000 and u = 2000 kbps:

u_s= 30 Mbps = 30000 Kbps

F = 15 Gbits = 15000 Mbits = 15000000 Kbps

D_{min} = 2 Mbps = 2000 Kbps

u_i = 2000/1000 = 2 Mbps

D_{cs} = max{1000*15000/30 , 15000/2} = max{500000,7500} = 500000 sec

D_{P2P} = max{15000/30 , 15000/2, 1000*15000/(30 + 1000*2)} = max{500,7500,7389} = 7500 sec

Client-server distribution (s)

Uploadspeed \n values	*n= 10*	*n= 100*	*n= 1000*
300 Kbps	*7500*	*50000*	*500000*
700 Kbps	*7500*	*50000*	*500000*
2 Mbps	*7500*	*50000*	*500000*

P2P distribution (s)

Uploadspeed \n values	*n= 10*	*n= 100*	*n= 1000*
300 Kbps	*7500*	*25000*	*45454*
700 Kbps	*7500*	*15000*	*20548*
2 Mbps	*7500*	*7500*	*7500*

5) (10 points) UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums) Show all work!

ANSWER:

0 1 0 1 0 0 1 1

0 1 1 0 0 1 1 0

\+

---

1 0 1 1 1 0 0 1

0 1 1 1 0 1 0 0

\+

---

1 0 0 1 0 1 1 0 1

Wrap around the overflow (Since we are considering 8 bits, the 9th bit to the left must be wrapped.)

0 0 1 0 1 1 0 1

0 0 0 0 0 0 0 1

+

---

0 0 1 0 1 1 1 0

Invert all bits to get 1s complement of the sum => 1 1 0 1 0 0 0 1

6) (10 points) A TCP machine is sending full windows of 65,535 bytes over a 1-Gbps channel that has a 10-msec one-way delay. What is the maximum throughput achievable? What is the line efficiency?

ANSWER:

65,535 bytes is 65535*8 = 524,280 bits.

1 Gbps is $1*10^9$ bits per second which is our bandwidth.

RTT is 10 msec * 2 = 20 msec which is 0.02 seconds since it is one-way (We also need to include ack.).

İn 20 ms RTT window, we can transfer $1*10^9 * 20/10^3$ = 20000000 bits of data which is $20*10^6$ that means we can send 20 mb of data in 20 ms at most. (TP is the maximum amount of data that can be relayed by the network within a unit time but we are calculating RTT window in this case.)

In question, it says Windows of 65635 bytes which is 524280 bits.

To achieve line efficiency (Throughput/bandwidth), we need to divide our window's that given in question to maximum we can send. So, 524280/20000000 = 0.026214 => Efficiency is 0.026214 * 100 percent = **2.6214%**

Throughput = efficiency * BW = 0.026214 * 10^9 = 26214000 bits per second = 26214 kbps = 26.214 mbps.

(Also, 26214000/50 = 524280 bits per window.)