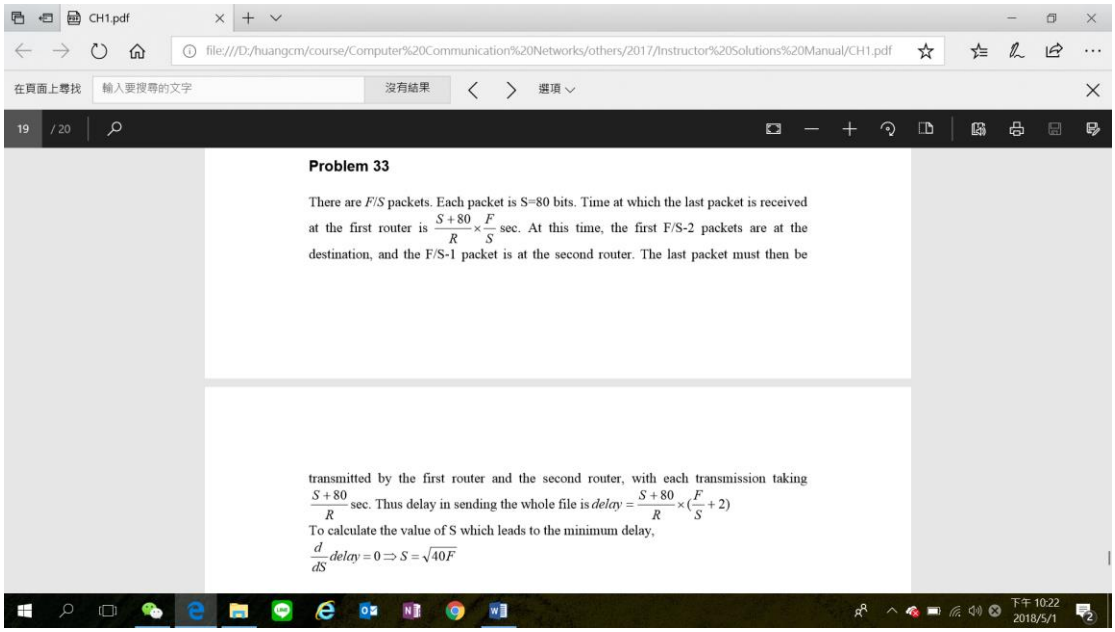
Computer Communication Networks

2021/Spring – Midterm Exam

1

A reliable data link protocol can recover from errors affecting transmissions on individual links. However, there are more sources of losses in an interconnection network formed by multiple routers. For instance, messages can be dropped by a router whose buffer is completely filled with packets. Moreover, a router may malfunction or break, thus destroying all buffered messages. Thus, it makes more sense to use a transport protocol for reliable data delivery in the considered scenario.

2



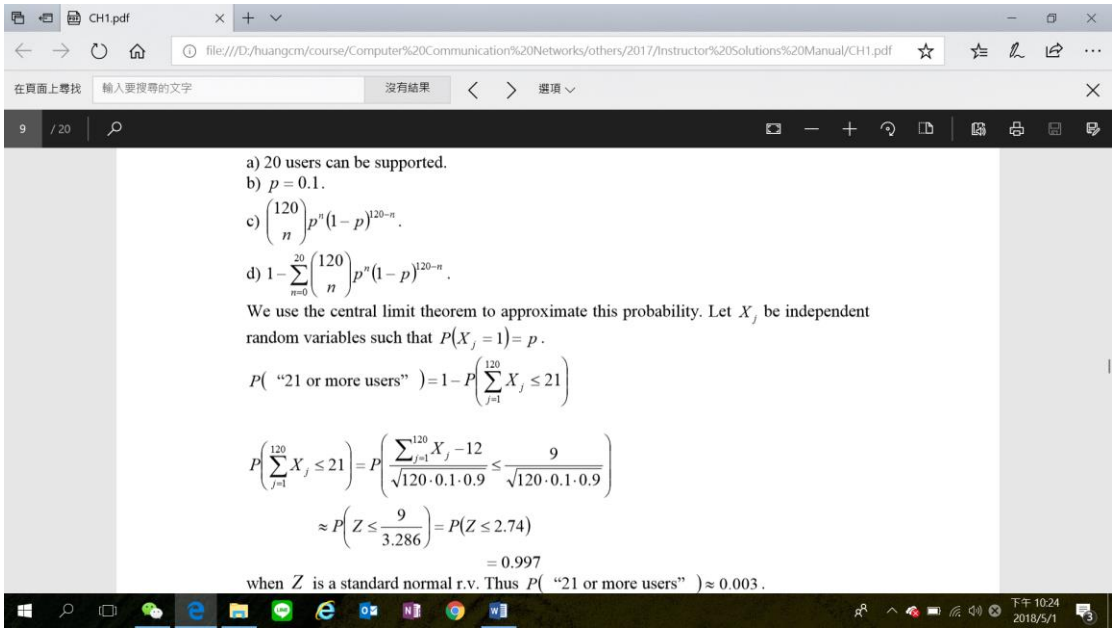
Problem 33

There are F/S packets. Each packet is S=80 bits. Time at which the last packet is received at the first router is $\frac{S+80}{R}\times\frac{F}{S}$ sec. At this time, the first F/S-2 packets are at the destination, and the F/S-1 packet is at the second router. The last packet must then be transmitted by the first router and the second router, with each transmission taking $\frac{S+80}{R}$ sec. Thus delay in sending the whole file is $delay=\frac{S+80}{R}\times(\frac{F}{S}+2)$

To calculate the value of S which leads to the minimum delay,

$\frac{d}{dS}delay=0\Rightarrow S=\sqrt{40F}$

3



a) 20 users can be supported.

b) $p=0.1$.

c) $\binom{120}{n}p^n(1-p)^{120-n}$.

d) $1-\sum_{n=0}^{20}\binom{120}{n}p^n(1-p)^{120-n}$.

We use the central limit theorem to approximate this probability. Let X_j be independent random variables such that $P(X_j=1)=p$.

$$P(\text{“21 or more users”})=1-P\left(\sum_{j=1}^{120}X_j\le 21\right)$$

$$P\left(\sum_{j=1}^{120}X_j\le 21\right)=P\left(\frac{\sum_{j=1}^{120}X_j-12}{\sqrt{120\cdot 0.1\cdot 0.9}}\le\frac{9}{\sqrt{120\cdot 0.1\cdot 0.9}}\right)$$

$$\approx P\left(Z\le\frac{9}{3.286}\right)=P(Z\le 2.74)$$

$$=0.997$$

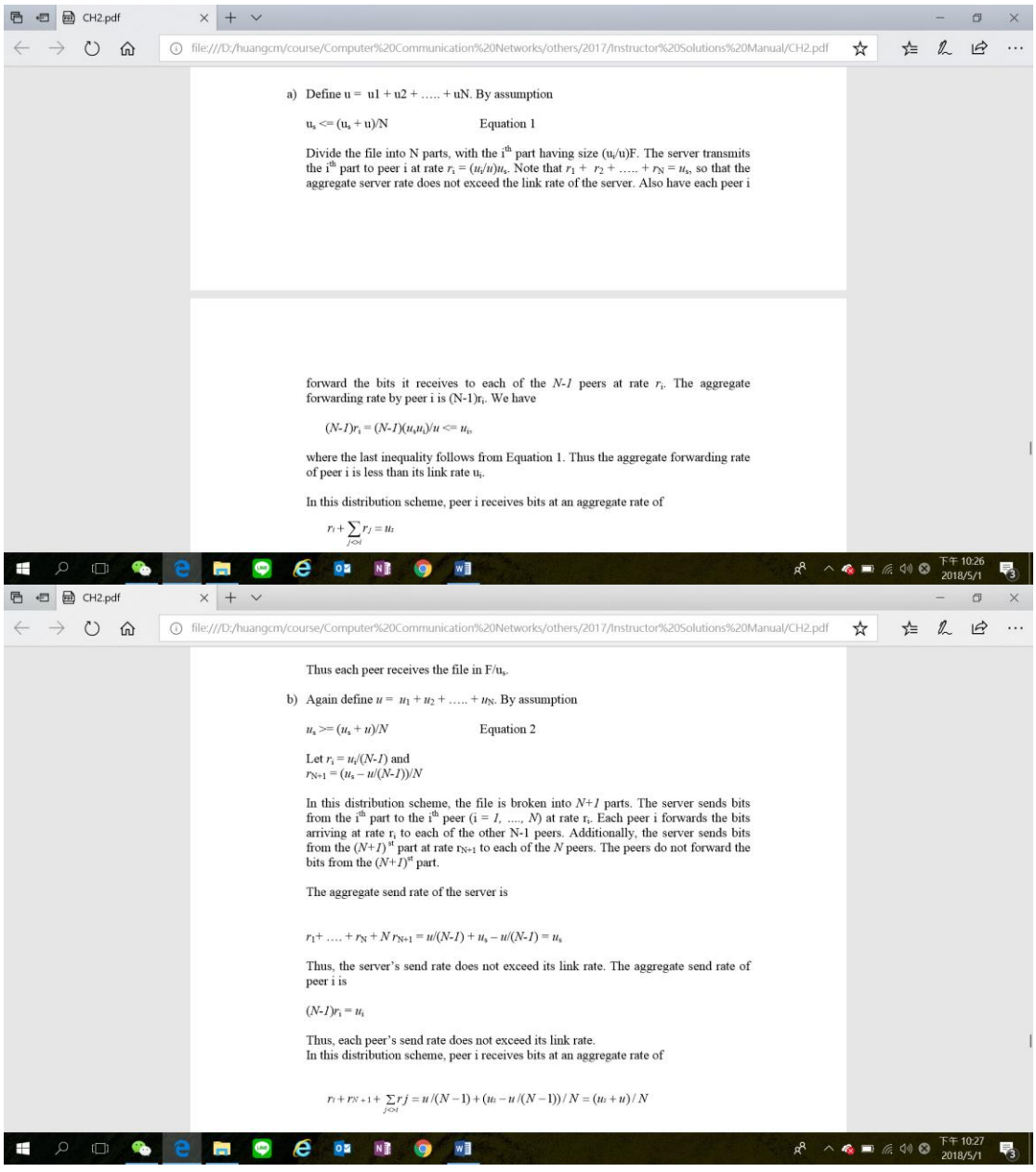
when Z is a standard normal r.v. Thus $P(\text{“21 or more users”})\approx 0.003$.

4

i) Recall that in BitTorrent, a peer picks a random peer and optimistically unchokes the peer for a short period of time. Therefore, Alice will eventually be optimistically unchoked by one of her neighbors, during which time she will receive chunks from that neighbor.

ii) For the TCP application, as soon as the client is executed, it attempts to initiate a TCP connection with the server. If the TCP server is not running, then the client will fail to make a connection. For the UDP application, the client does not initiate connections (or attempt to communicate with the UDP server) immediately upon execution

5



CH2.pdf

file:///D:/huangcm/course/Computer%20Communication%20Networks/others/2017/Instructor%20Solutions%20Manual/CH2.pdf

a) Define u = u1 + u2 + ... + uN. By assumption

$u_s <= (u_s + u)/N$ Equation 1

Divide the file into N parts, with the i^{th} part having size $(u_i/u)F$. The server transmits the i^{th} part to peer i at rate $r_i = (u_i/u)u_s$. Note that $r_1 + r_2 + ... + r_N = u_s$, so that the aggregate server rate does not exceed the link rate of the server. Also have each peer i

forward the bits it receives to each of the *N-1* peers at rate r_i. The aggregate forwarding rate by peer i is $(N-1)r_i$. We have

$(N-1)r_i = (N-1)(u_s u_i)/u <= u_i$,

where the last inequality follows from Equation 1. Thus the aggregate forwarding rate of peer i is less than its link rate u_i.

In this distribution scheme, peer i receives bits at an aggregate rate of

$r_i + \sum_{j<>i} r_j = u_i$

下午 10:26
2018/5/1

CH2.pdf

file:///D:/huangcm/course/Computer%20Communication%20Networks/others/2017/Instructor%20Solutions%20Manual/CH2.pdf

Thus each peer receives the file in F/u_s.

b) Again define $u = u_1 + u_2 + ... + u_N$. By assumption

$u_s >= (u_s + u)/N$ Equation 2

Let $r_i = u_i/(N-1)$ and
$r_{N+1} = (u_s - u/(N-1))/N$

In this distribution scheme, the file is broken into *N+1* parts. The server sends bits from the i^{th} part to the i^{th} peer (i = *1, ..., N*) at rate r_i. Each peer i forwards the bits arriving at rate r_i to each of the other N-1 peers. Additionally, the server sends bits from the $(N+1)^{st}$ part at rate r_{N+1} to each of the *N* peers. The peers do not forward the bits from the $(N+1)^{st}$ part.

The aggregate send rate of the server is

$r_1 + ... + r_N + N r_{N+1} = u/(N-1) + u_s - u/(N-1) = u_s$

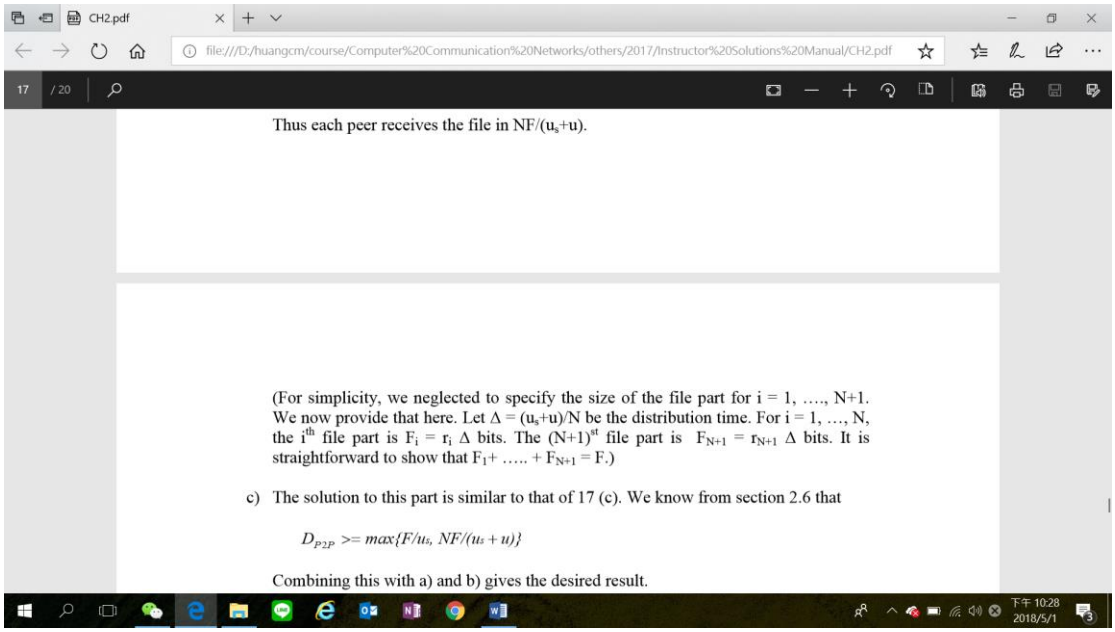
Thus, the server's send rate does not exceed its link rate. The aggregate send rate of peer i is

$(N-1)r_i = u_i$

Thus, each peer's send rate does not exceed its link rate.
In this distribution scheme, peer i receives bits at an aggregate rate of

$r_i + r_{N+1} + \sum_{j<>i} rj = u/(N-1) + (u_i - u/(N-1))/N = (u_i + u)/N$

下午 10:27
2018/5/1



6

i) N

ii) 2N

7

i) A UDP socket is fully identified by the destination IP address and the destination port. A TCP socket, instead, is fully identified by the source IP address, the source port, the destination address, and the destination port. This happens as TCP establishes a bi-directional full-duplex session between the sender and the receiver.

ii) Yes. The application developer can put reliable data transfer into the application layer protocol. This would require a significant amount of work and debugging, however.

8

No, the receiver cannot be absolutely certain that no bit errors have occurred. This is because of the manner in which the checksum for the packet is calculated. If the corresponding bits (that would be added together) of two 16-bit words in the packet were 0 and 1 then even if these get flipped to 1 and 0 respectively, the sum still remains the same. Hence, the 1s complement the receiver calculates will also be the same. This means the checksum will verify even if there was transmission error.

9

a) Here we have a window size of N=3. Suppose the receiver has received packet k-1, and has ACKed that and all other preceding packets. If all of these ACK's have been received by sender, then sender's window is **[k, k+N-1]**. Suppose next that none of the ACKs have been received at the sender. In this second case, the sender's window contains k-1 and the N packets up to and including k-1. The sender's window is thus **[k-N,k-1]**. By these arguments, the senders window is of size 3 and begins somewhere in the range [k-N,k].

b) If the receiver is waiting for packet k, then it has received (and ACKed) packet k-1 and the N-1 packets before that. If none of those N ACKs have been yet received by the sender, then ACK messages with values of [k-N,k-1] may still be propagating back.Because the sender has sent packets [k-N, k-1], it must be the case that the sender has already received an ACK for k-N-1. Once the receiver has sent an ACK for k-N-1 it will never send an ACK that is less that k-N-1. Thus the range of inflight ACK values can range from **k-N-1 to k-1**.

10)

Problem 26

There are $2^{32} = 4{,}294{,}967{,}296$ possible sequence numbers.

a) The sequence number does not increment by one with each segment. Rather, it increments by the number of bytes of data sent. So the size of the MSS is irrelevant -- the maximum size file that can be sent from A to B is simply the number of bytes representable by $2^{32} \approx 4.19$ Gbytes.

b) The number of segments is $\left\lceil \frac{2^{32}}{536} \right\rceil = 8{,}012{,}999$. 66 bytes of header get added to each segment giving a total of 528,857,934 bytes of header. The total number of bytes transmitted is $2^{32} + 528{,}857{,}934 = 4.824 \times 10^9$ bytes.
Thus it would take 249 seconds to transmit the file over a 155~Mbps link.