Name ______________________ Student No. ______________________

No aids allowed. Answer all questions on test paper. Use backs of sheets if necessary.

Total Marks: **30**

[10] 1. The article *Making the Web faster with HTTP 2.0* by Grigorik, mentions several new features of HTTP 2.0 such as:

(a) Stream Prioritization

(b) Server Push

(c) Header Compression

Pick two of them and explain what they are.

Solution: Stream Prioritization: Once an HTTP message can be split into many individual frames, the exact order in which the frames are interleaved and delivered within a connection can be optimized to further improve the performance of an application. Hence, the optional 31-bit priority value: 0 represents the highest-priority stream; $2^{31}-1$ represents the lowest-priority stream. Not all resources have equal priority when rendering a page in the browser: the HTML document is, of course, critical, as it contains the structure and references to other resources; CSS is required to create the visual rendering tree (you cannot paint pixels until you have the style-sheet rules); increasingly, JavaScript is also required to boot- strap the page; remaining resources such as images can be fetched with lower priority.

Server Push: the ability of the server to send multiple replies for a single client request — that is, in addition to the response to the original request, the server can push additional resources to the client without having the client explicitly request each one.

Header Compression: Each HTTP transfer carries a set of headers used to describe the transferred resource. In HTTP 1.x, this metadata is always sent as plaintext and typically adds anywhere from 500 to 800 bytes of overhead per request, and often much more if HTTP cookies are required. To reduce this overhead and improve performance, HTTP 2.0 compresses header metadata: Instead of retransmitting the same data on each request and response, HTTP 2.0 uses header tables on both the client and server to track and store previously sent header key-value pairs. Header tables persist for the entire HTTP 2.0 connection and are incrementally updated both by the client and server. Each new header key-value pair is either appended to the existing table or replaces a previous value in the table. As a result, both sides of the HTTP 2.0 connection know which headers have been sent, and their previous values, which allows a new set of headers to be coded as a simple difference from the previous set.

This is problem P13 in chp 1 of Kurose (pg 73)

[10] 2. (a) Suppose that N packets arrive simultaneously to a link at which no packets are currently being transmitted or queued. Each packet is of length L and the link has a transmission rate of R. What is the *average* queueing delay for the N packets?

(b) Now suppose that a batch of N such packets arrives at the link prediodically every LN/R seconds. What is the average queueing delay of a packet in this case?

Solution: The queueing delay is 0 for the first transmitted packet, L/R for the second transmitted packet, and generally, $(n-1)L/R$ for the n-th transmitted packet (as we saw in class). Thus, the average delay for the N packets is:

$$
\begin{aligned}
&(L/R + 2L/R + 3L/R + \cdots + (N-1)L/R)/N \\
=&L/(RN)\sum_{i=1}^{N-1} i \\
=&(L/(RN))N(N-1)/2 \\
=&(N-1)L/(2R)
\end{aligned}
$$

For the second question note that it takes LN/R seconds to transmit such a batch of N packets. Thus, the queue is empty each time a new batch arrives. Thus, the average delay of a packet across all batches is the average delay within one batch, which we computed in the first part to be $(N-1)L/2R$.

[10] 3. Consider distributing a file of $F = 15$Gbits to N peers. The capacities are as follows:

	upload	download
server	$u_s = 30$Mbps	—
client	u_c	$d_c = 2$Mbps

For each possible value $N = 10, 100, 1000$ and for each possible $u_c = 300, 700$Kbps and $u_c = 2$Mbps, prepare a chart giving the minimum distribution time for each of the combinations of N and u_c, for both Client-Server and P2P distribution schemes.

Justify your entries.

Client-Server	10	100	1000
300 Kbps	7680	51200	512000
700 Kbps	7680	51200	512000
2 Mbps	7680	51200	512000

P2P	10	100	1000
300 Kbps	7680	25904	47559
700 Kbps	7680	15616	21525
2 Mbps	7680	7680	7680

Solution: First of all, remember the formula for the Client-Server scheme:

$$D_{C-S} = \max\{NF/u_s, F/d_{\min}\}$$

and for the P2P scheme:

$$D_{P2P} = \max\{F/u_s, F/d_{\min}, NF/(u_s + Nu_c)\}.$$

Also, keep in mind that $F = 15$Gbits$= 15 \times 1024$Mbits, and that $d_{\min} = d_c = 2$Mbps. Now applying the formulas it is straightforward to obtain the numbers in the tables above.

Note that it is interesting that the better upload rate has no effect in the Client-Server paradigm, but provides a big improvement in the P2P paradigm.