Introduction to Computer Networks **Name:** ____________________
Fall 2018
Homework 1 (10/20/2018)

This homework contains 4 questions. The deadline is on Nov. 2 (Fri) at 23:59.
Please submit your answers (hard copy) to the TAs (office: EC-635).

---

1. (25 points) **Packet Switching and Circuit Switching:** Suppose users share a 1 Mbps link. Also suppose each user requires 200 kbps when transmitting, but each user transmits only 5 percent of the time.

   (a) (10 points) Explain briefly the difference between circuit switching and packet switching.

   (b) (5 points) When circuit switching is used, how many users can be supported?

   Solution:
   5

   (c) (5 points) When circuit switching is used, what percentage of the bandwidth, on average, is utilized.

   Solution:
   5%

   (d) (5 points) Assume users join the network uniformly randomly. When packet switching is used, ideally, at most how many users can be served by this link?

   Solution:
   $1/(0.2 * 0.05) = 100$

2. (30 points) **HTTP:** Assume that an HTTP servers only a single client directly through a single link of 1Mbps. Suppose that the server can support up to 5 TCP connections, each of which equally shares the bandwidth of the link between the client and the server. Consider a scenario where a client wants to download a webpage, which includes an HTML file of 100 Kbits and 10 images, each with the size of 200 Kbits. Assume that the round trip times (RTT) for building a TCP connection and sending an object downloading request are both 100 ms.

   (a) (10 points) Explain the difference between persistent HTTP and non-persistent HTTP.

Solution:
persistent: create a TCP connection for multiple object downloading
non-persistent: create a TCP connection for every object downloading

(b) (5 points) Assume the server can serve only a single TCP connection for every client. How many TCP connections the client should build if it uses persistent HTTP? How many TCP connections the client should build if it uses non-persistent HTTP?

Solution:
1 if persistent; 11 if non-persistent

(c) (5 points) Assume the server can serve only a single TCP connection for every client. What is the latency required to get all the objects if the client uses "sequential downloads" of *persistent HTTP*, i.e., creating only one TCP connection?

Solution:
Creating a single TCP connection: 100ms = 0.1s
Send request: 100ms * 11 = 1.1s
Download objects: (100,000 + 200,000*10) bits / 1,000,000 bps = 2100/1000 = 2.1s
Total = 0.1 + 1.1 + 2.1 = 3.3s

(d) (5 points) Assume the server can serve multiple parallel connections for every single client. What is the latency required to get all the objects if the client uses "parallel downloads" of *persistent HTTP*, i.e., creating up to 5 TCP connections simultaneously and each used to download multiple objects sequentially? (Hint: the client can only know how many objects it should download once it get the HTML file. **Note that** the link bandwidth should be shared by parallel connections. However, assume that the RTT will be the same even if multiple requests are sent simultaneously.)

Solution:
Create one TCP connection in the beginning and create the other four TCP connections in parallel after getting the HTML file: 100ms * 2 = 0.2s
Send one request for HTML, five requests in parallel for the first five objects, and then the other five requests in parallel for the remaining objects: 100ms * 3 = 0.3s
Download one HTML file, five objects in parallel, and then the other five objects in parallel: 100,000/1,000,000 + (200,000*2) bits / (1,000,000/5) bps = 0.1 + 400/200 = 2.1s

Total = 0.2 + 0.3 + 2.1 = 2.6s

(e) (5 points) Assume again the server can serve multiple parallel connections for every single client. What is the latency required to get all the objects if the client uses "parallel downloads" of *non-persistent HTTP*, i.e., creating up to 5 TCP connections simultaneously?

Solution:

Sequentially download the HTML file, five objects in parallel and the other five objects in parallel.

Sequentially creating one TCP connection, five TCP connections in parallel and the other five TCP connection in parallel : 100ms * 3 = 0.3s

Send one request for HTML, five requests in parallel for the first five objects, and then the other five requests in parallel for the remaining objects: 100ms * 3 = 0.3s

Download one HTML file, five objects in parallel, and then the other five objects in parallel: 100,000/1,000,000 + (200,000*2) bits / (1,000,000/5) bps = 0.1 + 400/200 = 2.1s

Total = 0.3 + 0.3 + 2.1 = 2.7s

3. (20 points) **Delay:** Consider two hosts, A and B, connected by a switch S. The link A↔S is 10 Mbps and has a propagation delay of 5 ms; the link B↔S is 100 Mbps and has a propagation delay of 20 ms.

(a) (10 points) Suppose A sends a single 100 Kbits packet to B. What is the total (end-to-end) delay for B to receive the packet entirely? Assume that S operates in a "store-and-forward" manner and that the processing delay is 0. Also, the switch S only serves A and B, i.e., no other users.

Solution:

Tx delay A-S: $0.1Mb/10Mbps = 0.01s = 10ms$
propagation delay A-S: $5ms$
Tx delay S-B: $0.1Mb/100Mbps = 0.001s = 1ms$
propagation delay S-B: $20ms$
total: $10 + 5 + 1 + 20 = 36ms$

(b) (10 points) Suppose A sends "two" 100 Kbits packet back-to-back to B. What is the total (end-to-end) delay for B to receive the two packet entirely?

Solution:

Tx delay A-S: $0.1Mb/10Mbps = 0.01s = 10ms$
propagation delay A-S: $5ms$

Tx delay S-B: $0.1Mb/100Mbps = 0.001s = 1ms$
propagation delay S-B: $20ms$

The second packet needs 10 ms to be sent by A. During this 10 ms, the first packet has arrived S and be forwarded by S (taking $5 + 1 = 6ms$). Hence, the second packet does not need to wait at the switch. Since it only introduces an additional 10 ms delay from A, the overall E2E delay equals $36ms + 10ms = 46ms$.

4. (25 points) **DNS.**

(a) (10 points) Explain why DNS is a "distributed" and "hierarchical" design.

Solution:
distributed: multiple servers share the loading
hierarchical: different servers have different tasks, e.g., root server, authoritative server and local server

(b) (10 points) Compare the difference between "iterative query" and "recursive query".

Solution:
iterative: the server who receives the request only gives the information of the next server, but not forward the request
recursive: the server who receives the request directly forwards the request to the next server

(c) (5 points) What are the IP addresses of NCTU's DNS servers? Go to search in the website of NCTU computer centers, instead of using the IPs given in the slide. (Not sure whether they have been changed.) Explain where (or how) to you find your answer.

Solution: 140.113.1.1, 140.113.250.135, 140.113.6.2
as shown in http://www.it.nctu.edu.tw/?page_id=89&lang=en