CSC358 Tutorial 2

Question 1: Concept Review

(a) While writing software at the application layer, do we need worry about the transport-layer protocols? Do we need to worry about the network-layer protocols? Why or why not?

(b) What do we mean by "HTTP is stateless"?

(c) What do we mean by "Cookies enable states"?

(d) What's the difference between persistent and non-persistent HTTP?

(e) What are the pros and cons of web cache (proxy)?

Question 2: Web Cache Response Time

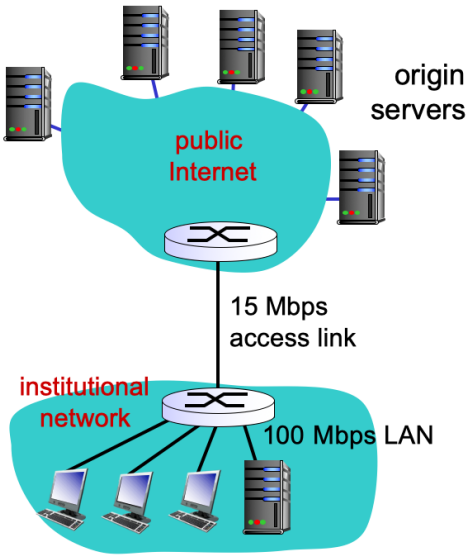
Consider the network in the picture below. There is an institutional network connected to the public Internet through a 15 Mbps access link. Suppose that the average object size is 850 Kbits and that the average request rate from the institution's browser to the origin servers is 16 requests per second. We model the total average response time as the sum of the *average access delay* (i.e. the delay from Internet router to institution router) and the *average Internet delay.* For the average access delay, use the following formula:

$$\text{average access delay} = \frac{\alpha}{1 - \alpha \cdot \beta}$$

where α is the average time required to send an object over the access link and β is the arrival rate of objects to the access link. The *Internet delay* is the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response. We assume the Internet delay is 3 seconds on average.

(a) Find the total average response time given the above configuration (without a web cache).

(b) Now suppose a web cache is installed in the institutional LAN. Suppose the cache hit rate is 0.6. Find the total average response time again.



Question 3: Persistent/Non-Persistent HTTP and Parallel Downloads

Consider a link over which a host can transmit at a rate of 150 bps in both directions. Suppose packets containing only *control* (e.g., HTTP request header, handshaking for setting up the connection) are 200 bits long, and packets containing *data* are 100 Kbits long (including both header and body of the HTTP response). Assume that if a host uses N parallel connections then each gets $1/N$ of the link bandwidth. Now consider the HTTP protocol, and suppose that each downloaded object is 100 Kbits long (including both header and body of the HTTP response), and that the initial downloaded object contains 10 referenced objects from the same host.

(a) Using non-persistent HTTP without parallel connections, what's the total time taken to download all objects?

(b) How about using persistent HTTP without parallel connections?

(c) How about using non-persistent HTTP with 10 parallel connections?

(d) Comparing the strategies in (b) and (c), which one allows faster download of all objects? Is the difference significant?

(e) Suppose that the link is shared by Alice with four other users. Alice uses parallel instances of non-persistent HTTP, and the other four users use non-persistent HTTP without parallel downloads. Do Alice's parallel connections help her get web pages more quickly? Explain your answer.

(f) If the other four users now start opening parallel instances of non-persistent HTTP, then would Alice's parallel connections still be beneficial? Explain your answer.

(g) Now take the *propagation delay* into account and redo the analysis in Part (d). Assume that the propagation speed is 2×10^8 m/sec. Do some calculations and discuss.

(h) Suppose you are developing a web browser, should you implement parallel connections or not? Have a discussion.