Name: AndrewId:

15-441 (Fall 2012)
Question Set #3

1. Consider a sliding window protocol, such as the one we discussed in class. What effect does the window size have upon throughput? Please derive a formula that expresses the maximum *throughput* (bits/second) of the connection between two systems as a function of the *bit rate* of the channel, the one-way *latency* of the channel (seconds), the *window size* (number of frames), and the *frame size* (bits).

2. Some applications have a natural tendency to emit a large number of small network rights. If improperly managed, these result in a large number of small packets, each of which bears the full overhead of all of the packet metadata at each level of the protocol stack. The result is a dramatically inefficient use of network time.

    a. One solution to this problem is simply to aggregate many small writes into larger writes. But, this solution causes performance problems for certain common classes of applications. Please identify the common classes of applications and provide three specific examples of situations where delaying and aggregating small writes affects them.

    b. Why does the delay-and-aggregate strategy cause problems in these cases?

3. TCP has several options for managing acknowledgments, among them are *delayed acknowledgment, selective acknowledgment, cumulative acknowledgment,* and *piggyback acknowledgment.*

    a. Why doesn't it make sense to acknowledge all packets immediately?

    b. What is hidden from the sender by simply cumulative acknowledgment that is made visible by selective acknowledgment? How can the sender act upon this information?

    c. Piggyback acknowledgment is designed to reduce the overhead of sending small acknowledgment messages. But, this strategy comes at a price. What is that price?

4. Recall that *Nagle's Algorithm* is a technique for reducing the overhead of small messages. Under some situations delayed acknowledgment and Nagle's Algorithm can interact badly. Please characterize this bad interaction and describe the situations under which it might occur

5. I suggested in class that TCP will generate a duplicate ACK if a segment being lost or reordered. Please draw a picture that demonstrates each of these situations.

6. We talked a lot about TCP's congestion control. What causes congestion?

7. Given the existing fabric of the standard internet protocols, how do routers communicate to senders that they have reached, or will soon reach, overburdened?

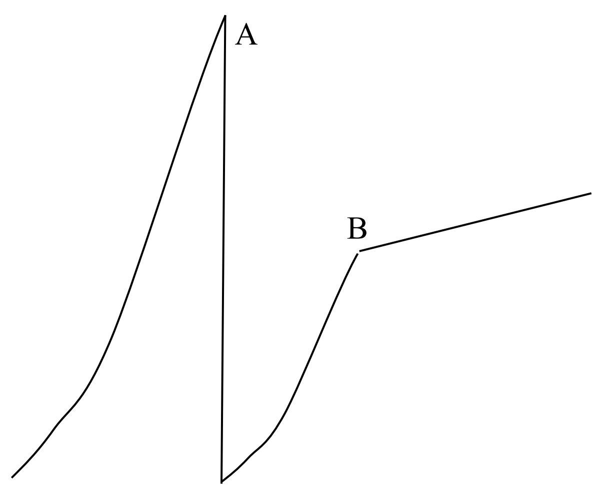
8. Please consider the plot below. It represents the congestion window size (vertical) against the transmission round/time (horizontal). Please identify each of the following, if present:

    a. The period(s) of slow-start.

    b. The period(s) of congestion control.

    c. The point(s) at which ssthresh is reached

    d. What happened at each of points A and B? Be specific: What are the two possibilities?



9. In HTTP v1.0, the server marked the end of the a transfer by closing the connection. This causes long latency in downloading most of the web pages. Why? Explain how Persistent HTTP addresses this problem.

10. In HTTP v1.0, the server marked the end of the a transfer by closing the connection. This also causes performance problems for the servers. Why? Explain how Persistent HTTP address this problem.

11. Is it possible that the same web object can cause multiple misses with the same proxy cache? If possible, explain the cause and provide an example or sequence of events. If not, please explain the constraints that make this impossible, and how they apply.

12. Consider the following experiment: You are opening up two FTP sessions from your laptop computer to a remote server. You are simultaneously getting a 3KB file and 10KB file from each of the sessions respectively. You repeat the experiments multiple times. While most of the times the 3KB file transfer finishes earlier, there are a few times that 10KB file transfer finishes earlier.

    Explain why this is possible given that these two transfers are between the same set of machines and at the same time.

13. *Jitter* is a common concern in the delivery of real-time media via the Internet.

    a. Intuitively, and without math, what is jitter?

    b. What are the causes of jitter?

    c. How and why does jitter affect our ability to utilize our available data carrying capacity ("bandwidth") in the case of real-time, interactive audio or video?

14. There is no one, agreed, formula for jitter. Methodfs for computing jitter range from simple, to the quite subtle. For our purposes, let's suppose that the jitter is the "average difference in latency between sequential packets". In other words, let's assume that three sequential packets have the latencies as follows:

    ```
    40ms, 52ms, 27ms, and 80ms
    ```

    Then we can calculate the jitter as follows:

    ```
    5.3mS = (|40ms-52ms)| + |(52ms-27ms)| + |(27ms-80ms)|)/3)
    ```

    Please perform a *traceroute* to your favorite **overseas** Website and compute the jitter based upon approximately 10 measurements. Please attach a copy of the traceroute output and your work to your homework solutions.

15. In the case where video or audio streams can be significantly buffered, why are latency and jitter less of a concern that the average bit rate ("effective bandwidth")?

16. The maximum data rate of a connection's poorest performing link can have a significant affect upon VoIP calls.

    a. Why is only the poorest performing link of the most significant concern?

    b. What steps can be taken, or tools can be used, to optimize a voice call for a less than generous data rate?

17. Consider NAT and firewalls. Why does Skype need so-called *supernodes*? And, what is their role in enabling communication between not-so-super nodes? Please illustrate your answer with pictures that show the role of super-nodes in communication between a super-node and a regular node, and two super-nodes. Be sure your answer speaks to the specific way in which NATs and firewalls function to enable certain communication, while restricting other communication.