# Homework 3: TCP

*Due: 11:59 PM, Dec 05, 2017*

## Contents

## Question 1

BBR gives us an interesting view of the relationship between window size, throughput, and round-trip time. Consider slide 19 of lecture 15 ('Another View of Congestion Control'):

1. As the window size increases past the BDP, why does the throughput stop increasing?
2. Assume you set the window to a fixed size, larger than the BDP, and smaller than the 'BDP+Bottleneck Queue' point in the graph. Explain why the queue at the bottleneck router is fixed at this point, instead of being ever growing.
3. Explain why you can't measure the minimum RTT ($RTT_{prop}$) and the bottleneck bandwidth at the same time. How does BRR get around that?

## Question 2

TCP Reno is not fair when two flows sharing the bottleneck link have very different round trip times. In the original Raj Jain diagram we used in class (e.g., slide 5 in lecture 15), we assumed the nodes were making decisions synchronously (i.e., with the same RTT). If A has double the RTT of B, draw a similar diagram, indicating what the final rates of A and B will be.

## Question 3

When requesting several objects to display a web page from the same server, HTTP gives the client and server many options, which you will compare here.

1. Assume that packets are 1500B, and let's ignore header sizes here. The bottleneck bandwidth of the path to the server is 120 Mbps = (10,000 packets/s) and the RTT is 0.1s. Assume that the initial TCP window is 1 packet. Considering slow start, **what is the smallest object the server has to transfer before its congestion window leaves slow start?** (Assume that the content can have sizes that are multiples of the packet size. Assume you leave slow start as soon as the size of your window is larger than the BDP).
2. One option for HTTP is to do several requests on the same connection, with `keep-alive`. What are two advantages of this keep-alive over the old one-connection-per-request in HTTP 1.0?
3. What extra advantage does pipelining give? Is it more advantageous (over just keep-alive) when the objects are larger or smaller?

4. Another option clients have is to have several parallel connections. If your metric for success is the total time to download all objects, give one situation in which parallel connections are better than pipelining, and one in which it is worse than pipelining.

## Question 4

TTL in DNS is quite different from TTL in IP.

(a) Give one advantage for a server to set a very high TTL in a DNS entry.

(b) Give one situation in which a server would want to set a very low TTL in a DNS entry.

(c) During an attack last year, Dyn, a company that was serving as the primary DNS provider for a number of other companies, had its authoritative DNS servers go down. After the TTL for the cached DNS responses for their clients expired (Twitter, for example, was affected), people couldn't them. One DNS resolver provider, OpenDNS, however, decided to disrespect the protocol and was returning entries for the affected companies even after the TTL expired. (They were doing this in cases in which the authoritative server for an expired entry could not be reached). While this let people access the affected sites, cite one potential disadvantage of this approach.

(d) CDNs use DNS responses to return good servers for you to find content with low latency. Explain why using a DNS resolver out on the Internet can result in poor choice of CDN servers.

(e) IP Anycast is one way to alleviate this problem. Explain how IP anycast is implemented through BGP. Explain how the use of anycast by such a DNS provider (e.g., Google's 8.8.8.8) can alleviate the problem in the previous question.

---

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS168 document by filling out the anonymous feedback form:

`https://piazza.com/brown/fall2017/cs168`.