Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.829 Fall 2020 **Problem Set 3** November 2, 2020

This problem set has two problems. Answer the written parts as clearly and concisely as possible. Follow the submission instructions for the coding part.

You may collaborate and discuss ideas with others in the class, **but your solutions and presentation must be your own**. You may work with a partner if you so choose on problem 2b only. Do not look at anyone else's solutions or copy them from anywhere. Please list your collaborators on your submission.

1 Traceroute, Routing, and CDNs (40%)

Useful resources for this task:
`https://bgp.he.net/` Map IP addresses to Autonomous Systems and prefixes (and more).
`https://www.peeringdb.com/` Information about Autonomous Systems, peering facilities, etc.
`http://as-rank.caida.org/` Ranking of Autonomous Systems based on their customer cone.

(a) Login into one of CSAILs machines, e.g., `login.csail.mit.edu`. Execute a DNS lookup using the command `dig` for `www.google.com` and `www.mit.edu`. In case that dig responds with multiple IP addresses (`A` records), choose one for each domain. Which Autonomous Systems (ASes) originate these IP addresses? Which provider does MIT use to host its public websites?

(b) Execute a `ping` command to both IP addresses. Based on the minimum RTT observed and speed-of-light constraints, what is the maximum possible geographic distance of these servers from Cambridge? Assume fiber-optic links (light in fiber travels at 2/3 of the speed of light c).

(c) From a CSAIL machine, execute a traceroute towards Google's resolved IP address. Use `traceroute -A <IP>` to resolve individual hops to AS numbers. Which ASes does the packet traverse from MIT's network towards Google? Inspect the router names on the traceroute path. Approximately where geographically do the packets enter Google's AS?

(d) Visit `https://www.telstra.net/cgi-bin/trace` in your browser. The Web interface allows you to execute traceroutes from a server in AS1221 in Melbourne, Australia. Execute a traceroute to `www.mit.edu` and `www.google.com` from this machine. The last hop on the traceroute corresponds to the server. Which ASes originate the two IP addresses? In the case of MIT's CDN provider, what does this reveal about their server deployment strategy? Based on the RTTs for the last hop on both traceroutes, in which area of the world, roughly, are the servers located?[1]

[1] if traceroute shows exceptionally high and variable RTTs, please re-run the traceroute command until you find a stable minimum RTT.

(e) From the CSAIL machine, execute a traceroute to both server IP addresses found in the traceroute executed from Australia. Which ASes do the packets traverse towards Australia? Inspect router names, AS numbers and RTTs on both traceroutes: Which ASes carry the packets over long geographic distances and in which geographic locations, approximately, do packets traverse between ASes?

(f) Do your measurements challenge the classical notion of a strictly hierarchical provider/customer Internet? Discuss briefly both the case of MIT's website (findings from (a) and (d)), as well as the case of Google (focusing on your findings from (e)).

(g) **Bonus:** Execute a traceroute from Cambridge to `www.naurugov.nr`. Inspect router names and RTTs along the path. What types of physical links do your packets likely traverse? Where is the destination IP address located?

(h) **Bonus:** From a CSAIL machine, execute a traceroute to `www.cloudflare.com`. Execute the same traceroute from Australia. Inspect the target IP address (last hop on the traceroute) and RTT values. What is happening here?

2 Adaptive Bitrate Selection (60%)

After her successful stint at Massive Internet Transfers, Alyssa P. Hacker was hired as CTO of Multinational Internet Television, a company that streams on-demand TV shows to users. The company is facing a critical problem: users on cellular connections have a terrible streaming experience: the variable link capacity means their video is either too low quality or they rebuffer too often. As her first hire, you've been tasked with implementing an ABR scheme to maximize the Quality of Experience for users on cellular connections.

You'll be designing and implementing your own ABR algorithms and evaluate them on a simulated setup.

Make a private repository, and initialize it with the starter code from
`https://github.mit.edu/jsalamy/6.829-pset-3-starter`.
Set up your environment according to the instructions in the repository.

1. Implement the Buffer-Based algorithm you read about for class. Scale all the traces so that their average throughput is 1Mbps (using the `network/scale_trace.py` utility) and run your buffer-based scheme on all of them. What's the distribution of average QoEs you see? Explain.

   Repeat the experiment for 2Mbps and 3Mbps and comment on any differences you see. What's the relationship between average QoE and average throughput of the trace?

2. Implement your own ABR algorithm, with the aim of beating the performance of the buffer-based approach. Your algorithm should use the average rate parameter provided to you in the `next_quality(..)` function. Write a report describing your approach and how it compares to the buffer-based scheme. **Note that the majority of marks for this part are for algorithm design and description, not leader-board performance.**

   We will host a leader-board for part (2) submissions. Details are in the repository.