Checkpoint 7: measuring the Internet

Due: end of class (June 7, 5 p.m.)

0 Collaboration Policy

Collaboration Policy: Same as checkpoint 0. Please fully disclose any collaborators or any gray areas in your writeup—disclosure is the best policy.

1 Overview

By this point in the class, you've implemented a significant portion of the Internet's infrastructure. This checkpoint is instead about **measuring** the actual Internet and reporting long-term statistics of a particular path.

This checkpoint is optional: If you're happy with your subjective scores on at least five of your earlier checkpoints, *you don't need to do this one.* But we still encourage you to do it—it's fun and you will probably learn something!

2 Collecting data

1. Choose a remote host on the Internet that has a RTT (from your VM) of at least 100 milliseconds. Some possibilities:
   - `www.cs.ox.ac.uk` (Oxford University CS department webserver, United Kingdom)
   - `162.105.253.58` (Computer Center of Peking University, China)
   - `www.canterbury.ac.nz` (University of Canterbury webserver, New Zealand)
   - `41.186.255.86` (MTN Rwanda)
   - *preferred:* an original choice with an RTT of at least 100 ms from you
2. Use the `mtr` or `traceroute` commands to trace the route between your VM and this host.
3. Run a `ping` **for at least two hours** to collect data on this Internet path. Use a command like `ping -D -n -i 0.2 hostname | tee data.txt` to save the data in the "data.txt" file. (The `-D` argument makes ping record the timestamp of every line, and `-i 0.2` makes it send one "echo request" ICMP message every 0.2 seconds. The `-n` argument makes it skip trying to use DNS to reverse-lookup the replying IP address to a hostname.)
4. Note: A default-sized ping every 0.2 seconds is fine, but please do not flood anybody with traffic faster than this.

3 Analyzing data

If you sent five pings per second for two hours, you will have sent approximately 36,000 echo requests ($= 5 \times 3600 \times 2$), of which we expect the vast majority to have received a reply in the ping output. Using the programming language and graphing tools of your choice, please prepare a report (in PDF format) that contains at least the following information:

1. What was the overall delivery rate over the entire interval? In other words: how many echo replies were received, divided by how many echo requests were sent?
2. What was the longest consecutive string of successful pings (all replied-to in a row)?
3. What was the longest burst of losses (all **not** replied-to in a row)?
4. How independent or correlated is the event of "packet loss" over time? In other words:
   - Given that echo request #N received a reply, what is the probability that echo request #(N+1) was also successfully replied-to?
   - Given that echo request #N did **not** receive a reply, what is the probability that echo request #(N+1) was successfully replied-to?
   - How do these figures (the **conditional** delivery rates) compare with the overall "unconditional" packet delivery rate in the first question? How independent or bursty were the losses?
5. What was the minimum RTT seen over the entire interval? (This is probably a reasonable approximation of the true MinRTT...)
6. What was the maximum RTT seen over the entire interval?
7. Make a graph of the RTT as a function of time. Label the x-axis with the actual time of day (covering the 2+-hour period), and the y-axis should be the number of milliseconds of RTT.
8. Optional: Make a histogram of the distribution of RTTs observed. What rough shape is the distribution?
9. Optional: make a graph of the correlation between "RTT of ping #N" and "RTT of ping #N+1". The x-axis should be the number of milliseconds from the first RTT, and the y-axis should be the number of milliseconds from the second RTT. How correlated is the RTT over time?
10. What are your conclusions from the data? Did the network path behave the way you were expecting? What (if anything) surprised you from looking at the graphs and summary statistics?

Please submit your report as a PDF via Gradescope, and have a wonderful summer! If you liked this class, some great next classes include CS155 (Computer and Network Security), CS244 (Advanced Topics in Networking), and CS249i (The Modern Internet).