CS 485/ECE 440/CS 585 Lab 2 (proposal for Lab 3)

Due by 11:59pm on Tuesday, 26 October, as an e-mail to the instructor (jedcrandall@gmail.com). Please send only PDF files. Lab 2 is worth 100 points. In grading it, I'll keep in mind that you only had a week to do it so consider the 100 points to be the relative importance of having a good proposal and not the relative amount of work compared to Lab 1.

Lab 2 is a proposal for your final project (which will be called Lab 3 and be worth 200 points). Your goal should be to teach the rest of the class something that can't be found in the book or via Google. You'll do this by running some experiments.

Your project should be related to either TCP congestion control (or, roughly speaking, anything in Chapter 6 of the textbook), or routing. I encourage you to think about topics that are related to some societal issue, such as network neutrality, Internet censorship, online privacy, networking in rural areas, or applications of networking to things like agriculture or wildlife preservation or something. This is not a strict requirement, but it's something to keep in mind when you're brainstorming.

It doesn't need to be original, publishable research, but it should teach the rest of the class something interesting that we couldn't find in the book or online. You should have some kind of a research question. This can be a hypothesis, or it can be something as simple as, “how much does A affect B?”-where your answer to the research question is a graph and not necessarily a test of a scientific hypothesis.

You should be in touch with me constantly over the next week to make sure that the proposal you submit is one that I'll approve and that I can give you the resources you need. We can run probably 100 virtual machines on shasta, and I can dig up other machines that can run 100 or even more. Also consider using open source programs like Tor. PlanetLab is a thought, but I'm not sure how long it takes to get on there and set up experiments. It might be days, or weeks, I don't really know since I haven't used PlanetLab much. Our campus does have PlanetLab nodes. We have 8 laptops left that you can check out (you can check out one more to have two in your group to connect as a pair and do TCP stuff, or you can check out two, four, or all eight if no other groups need them). You're also free to use your own resources as long as you understand the relevant university policies. You should not do anything untoward on university networks, of course, like flood them with a bunch of traffic. I can maybe set you up with an account on a Linux box that sits on a 1.5 Mbps Qwest DSL line, where I don't mind running Tor nodes or doing other things that we probably shouldn't do on campus. I might have some wired and wireless switches and hubs and some cables around that you could use. If TCP Segmentation Offloading interests you, I can give you access to a NIC card that does that. If we have a compelling reason, we can even make small purchases by begging for student fee funds, like for USB sticks that are wireless adapters. I already bought 22 laptops with student fees, though, so you'll have to help me with the begging part.

Your proposal should be no longer than 1 page, with 11 point font and 1-inch margins. It must contain the following sections:

- **Introduction:** brief summary of the proposal.
- **Intellectual merit:** this is where you describe what makes the experiments you plan to do interesting and what the class will learn from it that we didn't already know from the book and can't just find with Google.

- **Broader impact:** this is where you relate the proposed experiments and the domain you're working in to the real world and say how it impacts society in some way.
- **Planned experimental methodology:** this is where you state your research question, and say exactly what you plan to do to answer that question. It need not be a falsifiable hypothesis, it can be a quantitative question that you plan to answer with a graph. In other words, maybe you already know that TCP/IP with a certain option is better than without, but you want to make some graphs to show how much better it is under several different conditions.
- **Necessary resources:** this is where you tell me what you need. You should have already consulted me before submitting your proposal and be sure that it's something I can provide. Be sure to say whether the resources you put here are resources you have access to already or resources I should find a way to provide for you.
- **Proposed timeline:** when do you plan to have things done? Assume that the deadline for Lab 3 will be December 6th. We'll have poster presentations sometime after that.
- **Consultants:** Here you should say who you talked to about what you propose to do other than me. At a minimum, you should consult both TAs for the class (Shuang and Dustin--during their office hours, in class, or via e-mail) and one other faculty member besides me.

I'm including a sample proposal below. Keep in mind that the writing for the proposal should be entirely your own, nothing should be cut and pasted without attribution to the original source. The same will be true of the final writeup (Lab 3). Also keep in mind that how much of what you said you will do actually gets done will be an explicit part of your grade for lab 3, so be careful not to promise too much. What I'm using as an example below was a real class project that turned into a paper, it was an ambitious class project by a very hard-working and talented student, and it took 2 years to get the measurements right and eventually turn it into a paper, so your proposal should probably be slightly less ambitious than what I have below. Talking to me over the next week is the best way to strike the right balance between proposing something trivial that is not worth a 200-point lab *vs.* promising too much.

The paper is here if you're interested:

http://www.cs.unm.edu/~crandall/icdcs2010.pdf

The censors in China stopped doing HTML response filtering about halfway through our measurements. They probably realized at the same time as we did that injecting RSTs into a TCP/IP connection that is in full swing is a lot harder than injecting RSTs for a GET request that is piggybacked on the ACK that is the third part of a three-way handshake (before TCP slow start even begins).

Hypothetical Proposal for CS 485/585 Class Project: Measuring the Impact of TCP/IP Flow Control and Congestion Control on HTML Response Filtering Based on Forged RSTs

Jane Doe, John Doe, and John Q. Public

Introduction: HTML GET request filtering based on keywords in China has been studied in the past by Clayton *et al.* and Crandall *et al.* A question that remains is whether or not HTML response filtering occurs, and whether the flow control, congestion control, and routing issues of an intercontinental TCP/IP connection creates difficulties for the censors to guess the sequence number for forged RSTs. We plan to use public HTTP proxies inside China to elicit HTML response filtering and characterize the RSTs that are forged. We will focus on China since they are the only country that performs censorship at the router level, most other countries use web proxies for Internet censorship but we are interested in forged RSTs.

Intellectual merit: GET request filtering occurs very early in a TCP/IP connection so guessing the sequence number and successfully reseting the connection is straightforward. HTML response filtering, however, occurs later in a TCP/IP connection when the bandwidth has ramped up and a large amount of packet loss, packet reordering, retransmissions, congestion control, and flow control is occurring. We plan to report to the class on how this affects the censor's ability to correctly guess the sequence number and successfully reset the connection when blacklisted words appear.

Broader impact: Since HTML response filtering would be a powerful censorship technique if it could made to work effectively, assessing its effectiveness will give us some indications of possible future trends in global Internet censorship.

Planned experimental methodology: Our research question is: *How does the behavior of TCP/IP at an intercontinental scale affect the censor's ability to successfully reset a connection?* We will create three web pages with a blacklisted keyword in them and with names that will not elicit GET request filtering (since we are interested only in HTML response filtering): 1.html where the blacklisted keyword appears in the beginning of the web page, 2.html where it appears in the middle, and 3.html in the end. The blacklisted keyword will be repeated twice, *i.e.*, "falunfalun" instead of "falun", to negate the possibility of TCP segmentation breaking the keyword up so that no RSTs are elicited. These three HTML files will be placed on a web server we control, shasta.cs.unm.edu. For every public web proxy that we can locate (we expect to test 2 per day for 10 days), we will execute a script that queries one of the 3 web pages every 5 minutes (to avoid the timeouts reported by Clayton *et al.*) for a little over 12 hours until we have 50 data points per keyword placement. We will use wget with proxy settings to use HTTP proxies inside China. To account for diurnal patterns we will always start each experiment at noon New Mexico time and ensure that it finishes around midnight. We will record full tcpdumps for all experiments. We will make graphs showing the ratios of successful vs. unsuccessful resets at the application layer (which can be determined by the return code from wget) for all probes in which we observed a RST at the packet level in the tcpdump.

Necessary resources: We will need root access to sandpond.cs.unm.edu, which we ask the instructor to provide. We will also need access to a daily list of open public web proxies, which we already have.

Proposed timeline: We expect to figure out the wget proxy configuration and write a Python script to request pages via wget and the specified proxy by November 1st. This script will record the application-layer results of each download attempt (the connection was reset or not). From November 1st through November 10th we will collect data by running the script and recording tcpdumps on shasta. From November 10th through November 20th we will develop a Python script to parse the tcpdumps and, based on the timestamp, match every packet-level RST with an experiment at the application level. From November 20th through December 1st we will analyze the data. From December 1st through December 6th we will focus on the writeup and poster presentation.

Consultants: We talked to TA #1 during office hours, who suggested using wget since it allows for a proxy configuration and is easy to script. We talked to TA #2 about our experimental methodology plan via email, who suggested that we be careful about diurnal patterns. We also talked to Prof. X, who teaches yoga in the physical education department, and gave us a link to a good web site for finding open web proxies around the world.