COS 116
The Computational Universe
Laboratory 9: Internet Structure and Congestion Control

You saw in lecture that the Internet is a loose, decentralized network of computers that communicate with each other using *TCP/IP* (Transmission Control Protocol / Internet Protocol). We also saw that IP breaks data transmissions into *packets.* The Internet should not be confused with the World Wide Web (WWW); in fact it existed for many years before the WWW. In everyday usage the term WWW is used as shorthand for all useful content that resides on the Internet, but it also has a technical meaning: it refers to web pages that were all created using languages such as HTML (used for your course blog). These languages allow web pages to have *hyperlinks* to each other, thus giving rise to a conceptual network or graph which we thought about earlier in the context of web search. This conceptual network resides on the Internet, since web pages are stored on *server computers* (such as www.princeton.edu), which use the Internet to communicate with each other. These servers use the *HTTP protocol*, which, like every other means of Internet communication, uses IP.

Today's lab will give us a peek inside the Internet. Of special interest will be the phenomenon described in the lecture as "*dependence on the kindness of strangers*." There is no centralized control or predetermined limit on the amount of data communication allowed to a particular user or computer. Nobody can stop us all from going online at the same time and begin downloading our favorite files. However, it is clear that this could result in chaos: the Internet would get overloaded with packets and, unable to deliver all of them, would drop most. This used to happen a lot in the 1980s. Ultimately, most makers of Internet software agreed upon an informal convention for sharing the resources of the network that has so far kept the Internet running smoothly. It is interesting to speculate—from political, social, and technological viewpoints—how much longer this informal agreement will continue to last. In the meantime, it provides us with a good case study of congestion issues in society and their solutions.

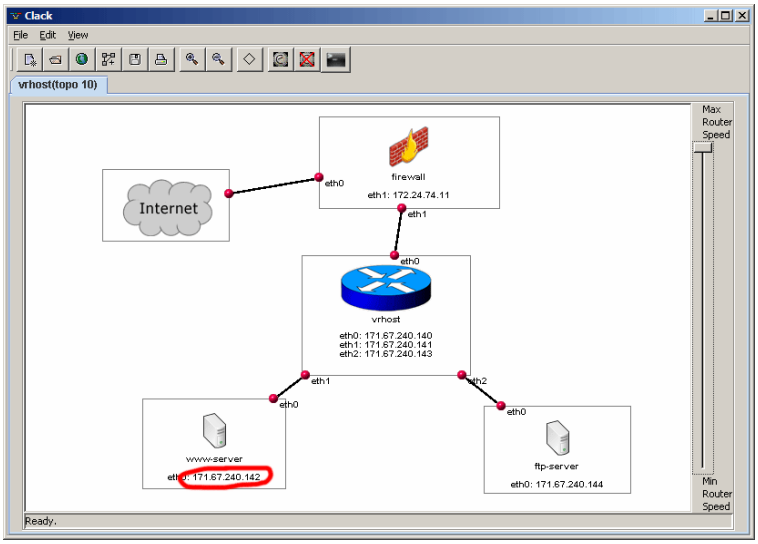
For this lab you'll be using Clack, a powerful network simulator tool created by Dan Wendlandt for his bachelor's thesis at Stanford. (Dan is now at Carnegie Mellon and has helped tremendously with this lab.) We have to resort to a simulator for experiments because the decentralized nature of the Internet makes it impossible to run controlled experiments. Clack creates a virtual world that merges seamlessly with the real Internet. Your browser thinks it is communicating with a server on the real Internet, but the requests from your browser are actually directed into a virtual world. Do not try to think too hard about where the real network ends and the virtual one begins, as that is not the point of this lab. (But, we are happy to answer any questions.) As usual, feel free to discuss with other students or a TA. However, you are not allowed to copy another student's answers.

Hand in your lab report at the beginning of lecture on Tuesday, April 17. Include responses to questions printed in bold. (Number them by Part and Step.)

Part 1: Entering the "Matrix"

Let us enter the virtual world of Clack.

1. **Start Internet Explorer and visit the following web page to launch Clack: http://www.clackrouter.net/princeton/lab.html**
   (If you are prompted with a security warning, click "Ok" and "Yes" as needed.)



2. **You should see the Clack application window, as pictured above.** Clack is set up to simulate a small network consisting of two servers ("www-server" and "ftp-server") connected to the Internet through a router ("vrhost") and a firewall. In this picture, you and your lab mates exist in the cloud labeled "Internet." We will refer to this picture as your *virtual world*.

3. **IP addresses:** Like all Internet devices, the ones in this virtual world are identified by numeric IP addresses consisting of four numbers separated by periods. As mentioned in lecture, each of the four numbers lies between 0 and 255. In the above picture we've circled the IP address of the virtual web server, 171.67.240.142. This address changes each time you restart Clack. Also, your neighbor is seeing a different IP address in his/her virtual world.

4. **Write down the IP address of www-server in your virtual world and mention it in your report.** (It probably will not be the same as the address in the picture.) In the instructions that follow, whenever we write *[www-server]* you should replace that with the IP address you wrote down in this step.

5. **Open a new Internet Explorer window and enter the address http://[www-server]/princeton/** If you watch the Clack window while the page loads, you should see some of the virtual network links flash green as data moves across them. Each flash corresponds to a single packet traversing the network.

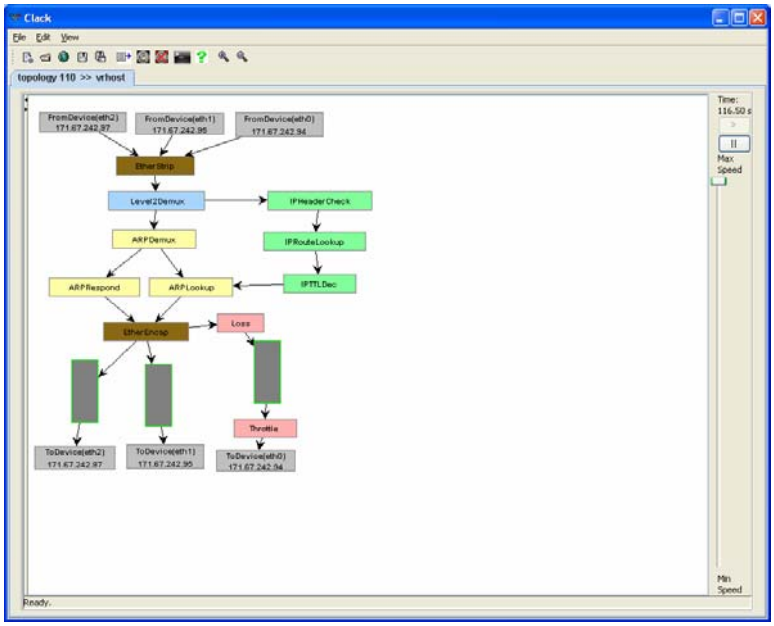
6. **Tell your neighbor your [www-server] address and ask him/her to open this address in their browser. Report what you see.** (We emphasize that this IP address is 100% real; feel free to call up mom on the cell phone and ask her to open this address in her browser. Do you see her request in your virtual world?)

7. **The web page you loaded from http://[www-server]/ princeton/ contains hyperlinks to several large files. Download one of these files such as "Dan's Magnum Opus". (Right-click on the link, select "Save target as…", and save the file anywhere.) Watch the download window for a minute, focusing on the transfer rate in kilobytes per second (kb/s). How does that number change? Stop the download and go on to Part 2, which will give you a glimpse into what is really going on.**

Part 2: Data flow and congestion

1. **Return to the web page at http://[www-server]/princeton/ and click again on the link labeled "Dan's Magnum Opus".** You will use Clack to observe TCP's congestion control behavior while this file is downloading.

2. **In the Clack window, double-click the blue icon labeled "vrhost" to enter the detailed view of the virtual world shown below:**



This screen shows a detailed view of how data moves through the simulated network. The window has more information than we will need, so ignore most of it. The important thing is that there are three entities attached to the network—a web server (eth1), an ftp server (eth2), and the "Internet" (eth0), which refers to you (and your mom, if she is participating). Data from these connections comes in at the gray boxes at the top of the screen and is directed to the appropriate destination gray box at the bottom of the screen. The file download that is currently underway is happening in the form of packets from the web server going from the gray box marked eth1 at the top to the gray box marked eth0 at the bottom.

3. **Identify the queues (vertical gray rectangles outlined in green near the bottom of the Clack display).** Physical data links on the Internet come with attached "queues." Queues are needed because packets may come in on a link faster than the device on the other end can receive them. So excess packets queue up to wait their turn to cross the link. The queue has finite capacity, and once it fills up any new incoming packets will be discarded ("dropped").

   The rightmost queue is on the network link leading to the Internet; it may flash green periodically as packets sent by the web server temporarily back up.

4. **Double click the rightmost queue. Using the controls that appear on the left side of the screen, set the Queue Size to 20,000 bytes.**

5. **Click the "Show Queue Size/Time Graph" button to display a graph of the queue's occupancy over time.**

6. **Back in the Clack window, double click the pink box labeled "Throttle." This lets you slow down the link between the web server and the Internet so that packets build up in the queue. Reduce the rate to 15%. After a little while, does the queue start flashing red?** This indicates that the queue is completely filled and packets are getting dropped.

   Return to the Queue Graph window and observe the occupancy of the queue over time. (**You may need to wait a little while before you can discern a pattern.)** When packets are dropped, the sender realizes this when it sees that the receiver has not acknowledged receipt of those packets. As soon as the sender realizes that any packet has been dropped, it halves its transmission rate. If packets continue being dropped after a while, it halves the transmission rate again. Once packets are no longer getting dropped, the sender begins to slowly increase its transmission rate to check out if the link can accept any more packets. It keeps increasing the rate until packets are dropped again, and then the whole process repeats. This results in a characteristic "saw tooth" pattern in the graph.

7. **Carefully sketch the graph and include it in your report. In your report, explain in a line or two why the curve is a saw tooth and stays that way. (After all, the flow has the link all to itself.)**

8. Think for a moment what will happen if a second flow now tries to use the link. The link is being fully used by a pre-existing flow. How will the two flows "share" this link? Note that the second flow has no idea who else is using the link nor how many of its packets will get through. It has to figure out the "right" amount. Try to predict the behavior. **To experimentally verify your guess, return to the page at http://[www-server]/princeton again and start a second download, the link labeled "Dan's Magnum Opus #2."**

9. **With both files downloading simultaneously, return to the Clack window, and in the ByteQueue control panel, click on "Clear Graph".**

10. **Now return to the Queue Graph window. Observe the two downloads for a couple of minutes (or long enough to establish their behavior). Carefully sketch the graph and include it in your report. Explain in a line or two what happened and why. Did the two flows settle into a fair sharing arrangement?**

Part 3: Congestion and Fairness in Other Protocols

Unfortunately, today's applications cannot always afford to behave as "good citizens." This experiment explores why this is so and what the consequences are.

1. **Cancel any downloads remaining from Part 1 and set the rate back up to 100% in the Throttle control panel. Also, set the Queue Size to 10,000 bytes in the ByteQueue control panel.**

2. Recall from Lab 4 that an audio stream must have a minimal bit rate (determined by the sampling frequency), below which the sound becomes severely degraded. **Suppose you are listening to a streaming MP3. If the stream has a bit rate of 16,000 bytes/second and it is transmitted using 1500 byte packets, on average, how many packets must be transferred every second to play the stream?** What if a packet gets dropped? The player could request a retransmission but that packet may take a while to arrive. **How do you think the player would respond to a dropped packet?**

3. **Launch QuickTime Player from the Start menu. On the File menu, select Open URL and enter the following location: rtsp://[www-server]/criminal.mp4**

4. **Wait for the music to begin playing and listen to it for a few seconds. Verify that it plays without skipping.**

5. **In the Clack window, double click the pink box labeled "Loss." In the control panel on the left, click the "Drop Next Packet" button a few times. Each time you click the button, a packet of the audio stream will be discarded on its way to your PC. Mention in your report what you heard. Experiment a bit and report how many packets had to be dropped and in how short an interval to hear this effect reliably. (You may need to drop packets somewhat aggressively before you hear an effect.)**

6. **Throttle the packet rate down to 15% using the Throttle control (this may cause the music to stop playing, though the server should keep transmitting the stream). Also, in the ByteQueue control panel, click "Clear Graph".**

7. **Open the Queue Graph and compare what you see now to what you saw in the graph of a single TCP download from Part 2. Observe the graph for long enough to establish its behavior. Carefully sketch the graph and include it in your report. How does the streaming server behave when it encounters congestion?**

   While web servers use the HTTP protocol, which operates on top of the TCP/IP protocol, the music file in this experiment is transmitted using a different protocol called RTSP (real-time streaming protocol), which operates on top of an alternative protocol called UDP. You will see this acronym in the graph. UDP isn't always a "good citizen." When a UDP application detects dropped packets, it can choose whether to reduce its transmission rate or whether to keep transmitting at the same rate.

8. **Throttle the packet rate back to 100% using the Throttle control. With the stream playing in QuickTime Player, open http://[www-server]/princeton again and begin downloading "Dan's Magnum Opus". Once the download has started, throttle the packet rate down to 15% again. Finally, in the ByteQueue control panel, click "Clear Graph".**

9. **Return to the Queue Graph and observe the behaviors of the two flows. The file download is marked "TCP" and the audio stream "UDP." Carefully sketch the graph and include it in your report. How does the behavior you see differ from the behavior of two simultaneous TCP downloads that you observed in Part 2? Does the UDP flow share fairly with the TCP flow? Explain the observation in a line or two.**

Part 4: Detective work

Now we use a program called *traceroute* (which you also saw in lecture) to do some sleuthing and discover the physical location of this virtual world. Recall that packets travel over the Internet by following a succession of intermediate links. Traceroute prints out this path.

1. **Open the Windows Start menu and click Run. Enter the filename "cmd" and click OK to open a command prompt.**

2. **At the command prompt, type "tracert cs.princeton.edu" and press Enter.** Each line in the trace lists one intermediary link that packets cross to get from your PC to the Princeton computer science department's network.

3. **Usually there will be more links in a trace to a distant network than to a nearby one, though this is not always the case. Trace the path to "cs.cornell.edu" and "cs.stanford.edu" and compare these to the Princeton trace. (Note: You may observe that some of the links are reported as "Request timed out". This is normal.)**

4. **The names of the intermediary links may appear cryptic, but they often contain useful hints about the physical location of the links. Looking closely at the Cornell trace, we can guess that the data travels through both Syracuse and New York, since one of the links is called "syr-7600-nyc-gsr.nysernet.net". Do you notice any major cities on the packets' path to Stanford?**

5. **Type "tracert [www-server]" at the command prompt and report what you saw. Where do you think the Clack virtual world physically resides?**

Further Questions

1. **Name a few Internet applications which you think might use UDP and briefly explain why.**

2. **The Internet is like the mail service in that it merely delivers packets; it does not keep track of who is sending packets to whom. (All bookkeeping, e.g., which packets were dropped, is done by sender and/or receiver.) This means in particular that a server cannot control who sends packets to it, though of course, it can just choose to not repond just as you choose not to respond to spam. Imagine you are a malicious attacker and want to prevent anyone from reaching Yahoo.com. Based on what you know about congestion control, how would you try to bring down the site? You can assume that the attacker has thousands of computers at his disposal.** (This is called a denial of service attack. An upcoming lecture on Internet Security will discuss it more, including how attackers gain control of thousands or millions of computers.)

3. **Why might Internet traffic from Princeton to Cornell travel by way of Syracuse, even thought Syracuse is not physically located between the two schools? Speculate based upon what you saw in lecture.**

4. **Mention two situations in everyday life that involve congestion. How does society deal with it (whether by law or by convention)?**