15-441: Computer Networks Spring 2010
Homework #4

Due: Apr 29^{th} 2010, in class
Lead TA: Kaushik Lakshminarayanan (`firstname@cs.cmu.edu`)

TCP Congestion Control

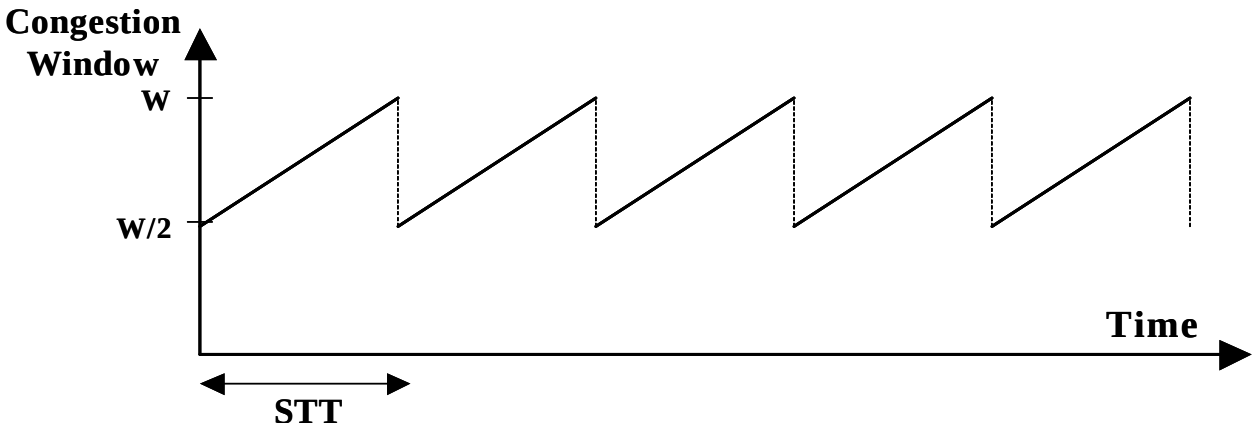


Figure 1: TCP sawtooth diagram

1. The picture above shows the famous TCP saw tooth behavior. We are assuming that fast retransmit and fast recovery always work, i.e. there are no timeouts and there is exactly one packet lost at the end of each "tooth". We are assuming that the flow control window is large and that the sender always has data to send, i.e. throughput will be determined by TCP congestion control.

   In the picture, W represents the congestion window size at which a congestion packet loss occurs (expressed in maximum transfer units). You can assume that W is large, so feel free to approximate (W-1) or (W+1) by W. STT represents the "saw tooth time" expressed in seconds.

   The aim of this exercise is to derive the average throughput of a TCP connection as a function of the roundtrip time (RTT), the maximum transfer unit (MTU), and the packet loss rate (PLR) for the connection. Please use the notation suggested by the figure, i.e. W and STT, as intermediate values if you need them.

   (a) Calculate the STT as a function of W, and the RTT. (Hint: Remember AIMD).

(b) How much data is sent in one STT?

(c) What is the average throughput of the connection?

(d) What is the average packet loss rate (number of losses per packet)?

(e) What is the relationship between the throughput and the packet loss rate?

QoS

2. A link shared by each of the clients listed in the table below is rate-limited by a token bucket with depth D = 10 packets and Rate R = 1 packet/sec (the shaping is per-client: each client gets its own D = 10, R = 1 and assume that the link is not the bottleneck, only the token bucket's rate-limiting is).

   (a) Each row in the table describes the sending behavior of a particular client over a 10 second period. Each column is a one second slot, where that client's entry in the column describes how many packets it sends during that slot. For each client, you must decide if the token bucket will allow all packets to be sent, or if the traffic will be limited. Circle either "allowed" or "limited" for each client in the far-right column of the table. If it is limited, indicate the time slot where it is first limited by circling the appropriate cell in the table. Assume the token bucket has 10 tokens at time 0.

Time	0	1	2	3	4	5	6	7	8	9	Result
client A	1	1	1	1	1	1	1	1	1	1	allowed / limited
client B	2	2	2	2	2	2	2	2	2	2	allowed / limited
client C	10	1	10	1	10	1	10	1	10	1	allowed / limited
client D	10	0	0	0	0	1	1	1	2	2	allowed / limited
client E	0	0	0	0	0	0	15	1	1	1	allowed / limited

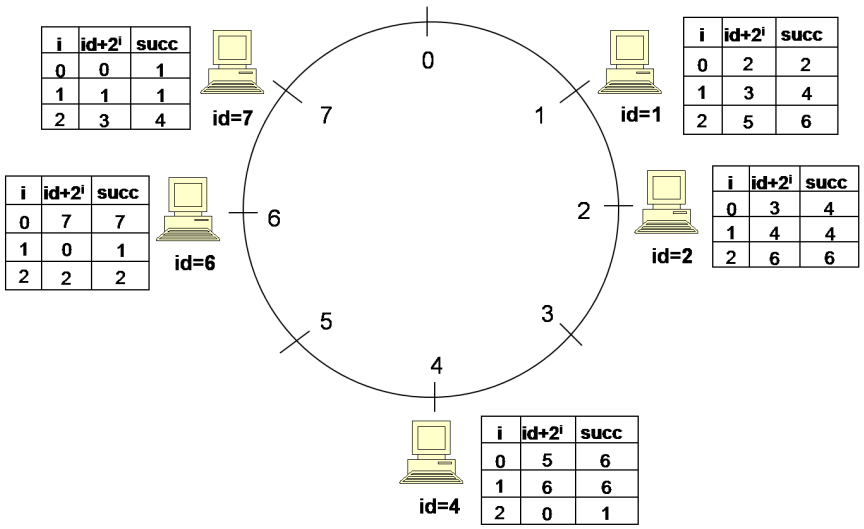
   (b) If the router that implements the token-bucket filtering above uses weighted fair queueing to share the link between its clients, what is the maximum delay through the router that any packets from any of the above flows will experience?

   (c) Which of the following two applications is more likely to benefit from a token bucket scheme compared to using a simple constant rate-limit? Circle your answer **and briefly explain why.**

   - HTTP requests from a web-browser
   - Voice-over-IP (VOIP) traffic from a Internet phone

P2P

3. Srini set up a P2P network to share poor jokes with other professors, but he's worried that the PJAA will shut down his centralized server, just like they did to Napster. So he set up a Chord ring for lookups and routing in his peer to peer network. Sadly, Srini's network is not popular, consisting of only four peers. The peers contain the listed have successor tables as shown (the $id+2^i$ column is there to remind you how the successor table is set up).



(a) Now, jokes (items) are added to the DHT. Which node(s) would store the jokes with the following *id*'s: 3, 5, 0, 2, 7, 1?

(b) List the nodes that will receive a query from *node 4* for *item 2*.

(c) List the nodes that will receive a query from *node 2* for *item 5*.

(d) Rui thinks that these jokes are awful, so he launches a DDoS attack and takes out node 4. Time passes, and the nodes converge on new routing tables that don't involve *Node 4*. Later, *Node 7* queries for *Item 5*. List the nodes that will receive this query.

Wireless

4. (a) Why is collision detection (as in Ethernet) not possible in wireless networks?

   (b) How does the sender identify collisions in wireless networks?

   (c) Why do hidden terminals cause collisions?

   (d) Which stage in the RTS-CTS exchange ensures that the data packet does not collide? Why? Why not the other one?

   (e) Consider two wireless standards A and B. B uses higher data rates also (using different modulation schemes) in addition to the modulation rates supported by A. Except for this difference, all other aspects of the standard B is same as A. Now, suppose CMU wants to support nodes having standard B also and plans on installing access points that support standard B (and A) so that the network can then support the higher rates of standard B. As a wireless expert who has done 15-441, you are being consulted regarding this deployment. What problems do you think the network might suffer (apart from the standard hidden and exposed terminal problems)? (Note: A node can hear another node only if it can decode the packet sent by the other node)

   (f) How would you deal with the above problems? Specifically, what modifications to the standard B do you suggest (if any) for inter-operability while having minimum performance degradation?

Security

5. Scenario: U.S. and Russia signed a comprehensive nuclear test-ban treaty. To verify the compliance of the treaty, seismic monitoring systems must be installed to detect any underground testing of nuclear weapons, Such monitoring systems are physically secure (tamper-proof) and are installed in the host nation near certain test sites. The data they collect is transmitted back to the monitoring nation (the host nation can also listen to the transmission), and this data will include timestamps to prevent any replay attacks. Let's assume in this problem the host nation is Russia, and the monitoring nation is the U.S.

   Requirements

   - The U.S. needs to ensure that the data it receives from the monitoring system is not altered/forged.
   - Russia wants to verify in real time that only the seismic data (agreed on in the treaty) is transmitted, i.e. the U.S. is not sending any additional information using the communication channel.
   - If the U.S. finds evidence (from the seismic data) that Russia violates the treaty, it wants to convince the United Nations that such a violation occurred.

   The following verification systems are developed to address the above requirements. (Assume that algorithms required for these approaches exist). We ask you to identify the problems with each of the four approaches. Note that there may be more than one problem with each approach. In this case, if you can identify one "important" problem with an approach, you will get full credit for that approach. If, however, you only identify several "less important" problems, you may not get the full credit. *Hint: Logical flaws in algorithms/protocols are "important", while other problems are "less important". Of course, if there are no logical flaws, other problems become "important".*

   (a) The U.S. installs a secret key along with the monitoring system. When transmitting data back to the U.S., the monitoring system encrypts the transmission using a symmetric cryptographic algorithm and the secret key. How does this approach address the requirements? What are the problems with this approach (if any)?

   (b) The U.S. generates a public/private key pair and installs the private key along with the monitoring system. The public key is made available to everyone. When transmitting data back to the U.S., the monitoring system signs the transmission with the private key using public-key cryptography (e.g. the RSA algorithm). How does this approach address the requirements? What are the problems with this approach (if any)?

(c) The U.S. and Russia each install a subkey along with the monitoring system (but they don't know each other's subkey). The monitoring system then constructs a private key from the two subkeys and also constructs the public key. The public key is made available to everyone, and public-key cryptography is used as in (b). How does this approach address the requirements? What are the problems with this approach (if any)?

(d) The monitoring system automatically generates a public/private key pair. The public key is made available to everyone, and public-key cryptography is used as in (b). How does this approach address the requirements? What are the problems with this approach (if any)?

Tools

6. In this section, you will learn a couple of practical tools: **ifconfig**, **netstat** and **tcpdump**. Below is a very brief description of what they do. For more detailed information, check the man pages (Note that on unix.andrew, tcpdump is located under /usr/sbin/. Either add this to your PATH or use the full path to run the commands). For the questions on tcpdump, please download the dump file from `http://www.cs.cmu.edu/~srini/15-441/S10/hw/hw4.dump`.

   netstat is a tool that can be used to display network connections, routing tables, interface statistics and many other things.

   tcpdump is a packet sniffer tool which can be used to get (and read) a trace of packets (traffic) received by the interface.

   (a) What does the command netstat -a show you? Explain the two parts of the output as well as the columns such as state, proto, type, etc.

   (b) What is the command to view the routing table of your machine using netstat? What is the command to only show IP addresses and not host names in the routing table?

   (c) How can you use netstat to find out what the network interfaces of your machine are?

   (d) What is the MTU of your Ethernet interface?

   (e) Identify the start and end of the TCP connection by giving the timestamps of the first packet and the last packet of the connection.

   (f) Give the timestamps of the SYN, SYN/ACK and the ACK packets of the TCP connection setup.

(g) What are the starting (initial) sequence numbers of the client and the server?

(h) What do the set of UDP packets immediately preceding the TCP connection correspond to?

(i) What do TCP/IP packets of length 52 bytes (66 bytes including Ethernet header) correspond to, in the trace?

(j) What acknowledgement scheme is used – cumulative or selective? Why?

(k) Given that the trace was taken on the node with IP address 128.2.209.192, can you identify the MAC address of the machine based on the given dump file alone? How?