6.033 Tcpdump

Kevin Cho
R5 Peter Szolovits 1PM

Due March 13th 2016

I Understanding tcpdump

1. The way we get the full IP address, we use the $-n$ parameter of tcpdump. The IP address of willow.csail.mit.edu is 128.30.4.222. The IP address of maple.csail.mit.edu is 128.30.4.223.

2. We can see that the port numbers are attached to the IP names when we don't use the $-n$ parameter. Thus, the port number of willow.csail.mit.edu is 39675. The port number of maple.csail.mit.edu is 5001.

3. The first communication was at 00:34:41.473036. The last one occured at 00:34:44.339015. This means that the time lapse was 2.865979 seconds. Now, we find the total kilobytes transferred by looking at the highest ack sequence. This value is 1572890. Thus, we had 1572.890 kilobytes transferred. Thus, the throughput was an average of $1572.890/2.865979 = 548.814$ kilobytes/second.

4. The time packet 1473:2921 was sent out at was 00:34:41.474225. However, the acknowledgement from the receiver came at the time 00:34:41.482047. This makes the RTT about 0.007822 seconds.

   For the packet 13057:14505, the time it was sent was at 00:34:41.474992. The acknowledgement came at 00:34:41.499373. This makes the RTT of the packet 0.024381 seconds.

   Now, the first packet's RTT was much smaller than the second's. This is because the queue buildup was much larger in the second than in the first. Meaning, because the first packet was relatively closer to the start, the queue buildup was not as large as when our second case of packets were sent. Thus, this causes our RTT to be different.

II Congestion Control

5. We hit packet losses when the graph has sudden drops in the number of outstanding packets. Because we start with a slow start, we find the max number of packets before the congestion starts. This number comes out to be about 750 outstanding packets. Thus, the machines can support about 750 packets before congestion/queue buildup starts.

6. The TCP started with a slow-start until about 0.5 seconds when it had a packet loss. There was also a slow-start at about 10 sec, and at about 16-17 seconds. The graph supports this by the sharp increase of outstanding packets until there was a packet loss at these three points in time. We know for the first, the TCP starts with a slow-start. For the second and third, it seems that the TCP continues with the slow-start and then decreases once there is a packet loss. It decreases to about 30-40 percent of packets before the loss. Then, it consists with the increase function until a packet loss occurs. Once this happens, we go through another slow start.

7. After each slow-start, there was a packet-loss. There was an additional packet loss before the second and third slow-start after the increase functions. The fast-retransmit/fast-recovery was in process after each slow-start because it decreased to about 30-40 percent of before the packet loss then proceeded with the increase function. Fast-recovery also seems to have occurred after the second increase function, but not after the first. This is because after the first, the packets were reduced to almost zero. After the second increase, the packets were only reduced to about 50 percent of what is used to be.

8. Yes, the graph would look different. The increase function is seen during the times of abot 1 sec to 10 sec, 11 sec to 16 sec, and 17 sec to the end of the graph. We see that the increase is actually a logarithmic growth instead of an additive increase. If it were an additive increase, the graph would not be curved, but it would be straight lines at these intervals. In addition, we would have seen much more staggering of up and downs with packet losses. Thus, during these intervals, we could have seen zig-zag lines.