CSC 458/D58/2209 Course Project
Due Date: Sunday, November 25, 2018

RTT Estimation in Real World

Introduction

Round Trip Time (RTT) is defined as the time that a packet needs to reach the destination and its acknowledgement is returned to the sender. We saw in the course that RTT is an important parameter of a connection and is required to tasks such as setting the timeout timer in TCP. In the course project, we have two goals.

First, we want to investigate how accurate and stable RTT estimation is in practice. There are different factors such as varying queueing delay and possible delay from receiver while it is waiting for new data to send and piggyback the acknowledgement on that data.

Second, we want to observe the difficulties of reconstructing the state of a TCP connection in the network. It is common to have stateful packet processing in the middleboxes. For example, in a stateful firewall, we may want to detect suspicious TCP connections. However, we simply see individual packets that belongs to a large number of different connections and we have to separate them and reorganize them as different flows.

We will use real-world packet traces to perform these tasks. Analyzing real-world packages also enables us to better understand properties of traffic that is usually traveling through our network.

Datasets

We will use the University 1 dataset that was collected and used in the famous IMC 2010 paper entitled “Network Traffic Characteristics of Data Centers in the Wild”. This dataset can be freely downloaded from here:

http://pages.cs.wisc.edu/~tbenson/IMC_DATA/univ1_trace.tgz

This is a compressed file. You can decompress is with the command “tar xzvf uni1_trace.tgz”. After decompression, you will notice that there are 20 packet traces in the archive: univ1_pt1 … univ1_pt20. Each group should only use one of these files. To find the trace file that you should use, use this formula:

trace_number = (student_number_1 + student_number_2) % 20 + 1

If you are working alone, use 0 for the second student number. For example, if the student number of the first person is 1234 and the student number of second person is 5678, then they should add these two numbers (1234+5678=6912), then find the reminder of dividing this number by 20 (6912 % 20 = 12) and add one to that (12+1=13). So this group should use packet trace number 13 (i.e., univ1_pt13).

Packet traces are in the pcap format. This is well known format for packet traces that is supported by many tools and there are many libraries to read/write them. Note that in these traces, the IP addresses are anonymized and packet contents are removed (i.e., you only see the packet headers in the trace). However, none of these limit our ability to perform the analysis that we want to perform.

There is not any specific requirement on how you will analyze the packet traces (e.g., which tools you are using or which programming language you use). This is up to you to select the best and most convenient tool(s) for this.

Deliverables

The main deliverable that you should submit is a PDF report that provides the answer to questions that we ask here and analysis of these results. You should submit this report as a single PDF with the name "report.pdf". You should clearly list any tools that you use. Furthermore, you should also submit any code/script that you wrote to perform the analyzes. You should compress all such files as a single archive named "code.zip".

We will use the MarkUs submission system that we used for PS1 and PS2 for the project report.

We want to remind everyone about the academic integrity. If you use any tool/resource/algorithm/code that is not developed by yourself, you should clearly cite the place that you take it from. Even when you wrote the code, if it is an implementation of someone else idea (e.g., someone else algorithm), you should cite it properly.

Required Analysis

In this section, we describe the analysis that you should perform and the required results that you should include in your report.

Dataset Statistics

This first analysis that you should perform is collecting general statistics about the traffic. This includes both per-packet and per-flow statistics. Note that for the size of a packet, you should check the header of the packet, because the payloads are not included in the packet traces and only headers are provided. For many of the CDF charts, it is better to use logarithmic scale for the x-axis. Explain why this is usually the case.

The per-packet statistics includes:

- The type of packet: in link layer (e.g., Ethernet), network layer (e.g., IP or ICMP), and transport layer (e.g., TCP, UDP). You should display these results as a table, both in terms of count and percentage of all packets. You should also include the total number of bytes for each.
- The size of packets: You should draw the CDF (Cumulative Distribution Function) of all packets, as well as TCP packets, UDP packets, IP packets, and non-IP packets. In case of IP, TCP, and UDP packets, also draw the CDF of packet header size. Analyze these results. For example, what difference you observe between TCP and UDP.

To analyze flows, you need to reconstruct flows. We only define TCP and UDP flows. We define a set of packets as a flow if they have the same source IP, destination IP, source port, destination port, protocol, and the maximum packet inter-arrival time between those packets is not more than 90 minutes. Note that you should combine both directions of each TCP or UDP flows together (otherwise, you will analyze each flow between A and B as two separate flows, one from A to B, and another one from B to A). The per-flow statistics include:

- Flow type: Similar to per-packet statistics, we want to know how many TCP and UDP packets we have. Report this result in a table similar to the one that you used for per-packet statistics.
- Flow duration: For each flow, measure the duration of the flow (the arrival time of the last packet minus the arrival time of the first packet). Then draw the CDF of flow durations for all flows, TCP flows, and UDP flows. Is there any difference between TCP and UDP flows?
- Flow size: Flow size can be measured both in terms of packet count as well as byte count. Measure the size of each flow (the number of packets in that flow, as well as the sum of size of all packets). Now draw the CDF of flow size (once using packet count and once using byte sum) for all flows, TCP flows, and UDP flows. Do you see any difference between TCP and UDP? What about the case of using packet count vs byte sum? For TCP packets, in the addition to the total byte sum, calculate the overhead ratio as the sum of all headers (including TCP, IP, and Ethernet) divided by the total size of the data that is transferred by the flow. If the flow did not transfer any data (e.g., the connection did not stablish successfully), use the number 9999 instead to represent infinity. Now draw the CDF of hit ratio. What can you say about TCP overhead base on this chart?
- Inter-packet arrival time: Calculate the arrival time between consecutive packets in each flow. Then draw the CDF of inter-arrival time between packets for all flows, TCP flows, and UDP flows. Is there any specific inter-arrival time that appears more commonly? If yes, is it present in all flows, TCP flows, or UDP flows? Do you see any difference between TCP and UDP flows?
- TCP State: For each TCP flow, determine the final state of the connection. You can do this by checking the TCP header flags. To simplify, assign each connection to one of these classes: Request (only the initial SYN packet), Reset (the connection terminated after one side sent a reset signal), Finished (the connection is successfully terminated after each side sent the FIN message and the other side acknowledged it), Ongoing (if the last packet was sent during the 5 minutes of the trace file), and Failed (if the last packets was sent before the last 5 minutes of the trace file and the connection is not terminated by a reset or FIN message. This case usually represents connections that suddenly disconnected, e.g., when one side is disconnected from the Internet).

RTT Estimation

Our second analysis focuses on RTT estimation. You should follow the formula and approach described in RFC 6298 for this. You can get this RFC from here:

https://tools.ietf.org/html/rfc6298

To perform this, one of the requirements is to map each acknowledgement to a packet. If you sent a packet multiple times (e.g., due to a packet loss), then you should not use them to estimate RTT (as it is mentioned in the RFC and we explained in the lectures). Considering that we are collecting the packets in the middle of the network, what we see will be the RTT between the point that we collect the data and the receiver. However, even with this limitation, we can still see any possible delay or variation in the RTT. Therefore, we will ignore this fact for this study. Just note that as a result of this, the RTT that we calculate when A sends to B will be the RTT between the collection point and B, and the RTT that we calculate when B sends to A will be the RTT between the collection point and A. Therefore, these two RTTs can be significantly different.

Now what we want to see if how RTT estimation is changing during each connection as well as during different connections between the same set of hosts. Considering that there are too many flows in the packet trace, we will focus on very specific flows to analyze this. For each of these flows, draw the sample RTT and estimated RTT as a function of time (as new packets arrive). Then analyze this chart. For example, is estimated RTT relatively stable? What about the sample RTTs? Do you see any increase or decrease in RTT? If yes, what can be the reason that the RTT is changing during the life of a connection?

You should select the following flows for this analysis:

- The top 3 largest TCP flows in terms of packet number
- The top 3 largest TCP flows in terms of total byte size
- The top 3 longest TCP flows in terms of duration

Next, we want to see how the RTT estimation by change from time to time between the same set of hosts. To do this, find the top 3 pair of hosts that have the highest number of TCP connections between themselves. For each of these host pairs, select all TCP flows between them. For each of those flows, calculate the estimated RTT base on RFC 6298 and then use the median estimated RTT as the representative RTT to represent this flow. Now draw chart of representative RTTs of different flows as a function of their start time (to demonstrate how these RTTs change over time) for this pair of hosts. Do you see any specific pattern? Is the representative RTT changes over time? Is this change random or is it following a specific patter (e.g., first increase, and then decrease)? Explain what could be the reason of this change. Also compare the 3 different host pairs together. Is there any difference between them? If yes, what could be the reason.