Alfred Tso Chun Fai 20012065g

COMP5311 Project: Performance Analysis of TCP and UDP

Introduction

Even wonder why Chrome browser feels so much faster than Firefox when watching Youtube videos ? I do and as a curious student i fired up Wireshark and capture some traffic and what i found is this.

9586	12:08:02.449306	142.250.66.110	10.0.0.23	QUIC
9587	12:08:02.449638	142.250.66.110	10.0.0.23	QUIC
9588	12:08:02.450505	142.250.66.110	10.0.0.23	QUIC
9593	12:08:02.452969	142.250.66.110	10.0.0.23	QUIC
9594	12:08:02.452971	142.250.66.110	10.0.0.23	QUIC
9595	12:08:02.453892	142.250.66.110	10.0.0.23	QUIC
9596	12:08:02.453894	142.250.66.110	10.0.0.23	QUIC
9597	12:08:02.453896	142.250.66.110	10.0.0.23	QUIC
9598	12:08:02.453897	142.250.66.110	10.0.0.23	QUIC
9599	12:08:02.454050	142.250.66.110	10.0.0.23	QUIC
9600	12:08:02.454052	142.250.66.110	10.0.0.23	QUIC
9601	12:08:02.454053	142.250.66.110	10.0.0.23	QUIC

According to wiki [^1], QUIC is used by more than half of all connections from the Chrome web browser to Google's servers.

What is QUIC

QUIC is a protocol aiming to improve web application that are currently using TCP [^2] by establishing a number of multiplexed

TCP v UDP: A case study

Background: Related Works

- Yuan et al. (2019) [ˆ3] compared the performance of TCP and UDP over SDN using average packet delay and packet loss probability. The study found that the TCP outperform UDP over the two metrics by 12-50% and 25-100% respectively. Note that here, we are not conducting our study in SDN.
- He et al. (2004) [ˆ4] studied the effect of, among other things, packet aggregation and ingress buffering on the throughput and delay jitter of TCP and UDP.
- Gamess et al. (2008) [ˆ5] proposed an upper bound model for TCP and UDP throughput in IPv4 and IPv6. The paper presented the model for calculating the maximum theoretical throughput of both protocol to be the ratio of the number of TCP or UDP payload bytes to transmit over the min. number of bytes necessary including IFG, Preamble, SFD, Ethernet DIX encapsulation, IP Header, TCP (or UDP) Header and Paddings, multiple by the bandwidth of the link, assuming full-duplex and no processing time.

Problem definition

Comparing transport layer protocol is no easy task, as there are many factors that could affect the result:

Different implementation

Some parameters within protocol (TCP for example) are left up to the implementation (which give rise to TCP/IP Fingerprinting, allowing malicious actor to identify, among other things, OS of the host) meaning that these difference could have subtle effect on the measurement of performance

Network Topology

The immediate route taken by the packets certainly would affect the result so the immediate routers, bottleneck links in between and other traffic could also affect the results.

The Bandwidth of the links involved

As mentioned above, we often might not be sure about the bandwidth of those immediate links or simply the link between client and the server. One of the paper mentioned made an assumption that the link would not have any collisions

Design/Methodology

Simulation or Real Lab

The first question of designing the said experiment is to simulate or not. Simulation might be better in our study as we can have more control over the considerations we mentioned. The biggest downside is how can we be sure that the result is “real” ? The answer to that is as long as we have a clearly defined scope for the experiment and frequent sanity check on these scopes, we can have an idea of how “real” the results can be.

Tool: NS-3

`ns-3` is a open-source, discrete-event network simulator used mainly for research or educational purposes written in C++. The simulator itself has a comprehensive documentation available online.

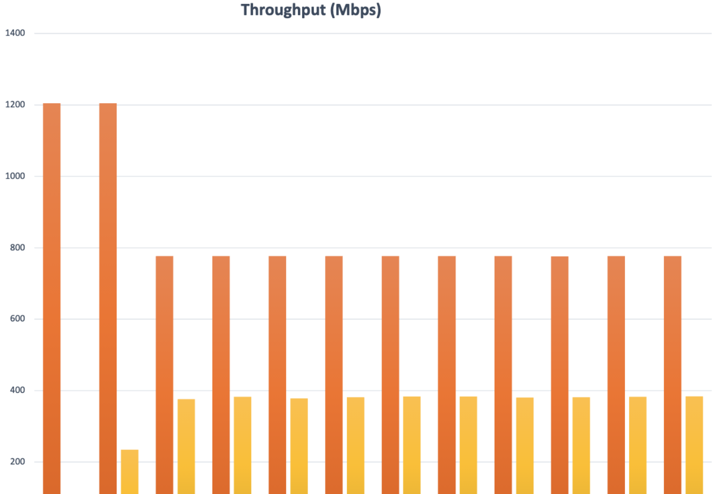
ns-3: basics

For our present purpose, we only need to understand the main abstractions of `ns-3`

Result

We will send 10, 100 and`range(1e4, 1e5, 1e4)` packets, each of 1472 Bytes to observe the flows of both TCP and UDP applications. Results are as follows:

Throughput



Discussion

Why not compare packets lost?

The `BulkSendApplication` made sure the once the lower layer send buffer is filled, it would wait. So the only place we could drop packets is in the channel which we will need to introduce error into. `UDPClient` has no such traffic control so we could observe the loss of packets.

Code

https://github.com/alfredtso/ns-3-project

Reference

ns-3 training resources QUIC

QUIC GoogleDoc

Yuan-Cheng Lai, Ahsan Ali, Md. Shohrab Hossain, Ying-Dar Lin,Performance modeling and analysis of TCP and UDP flows over software defined networks, Journal of Network and Computer Applications, Volume 130, 2019, Pages 76-88, ISSN 1084-8045

Jingyi He, S.-H.Gary Chan, TCP and UDP performance for Internet over optical packet-switched networks, Computer Networks, Volume 45, Issue 4, 2004, Pages 505-521, ISSN 1389-1286

Eric Gamess, Rina Surós, An upper bound model for TCP and UDP throughput in IPv4 and IPv6, Journal of Network and Computer Applications,Volume 31, Issue 4, 2008, Pages 585-602, ISSN 1084-8045

Shahrudin Awang Nor, Raaid Alubady, Wisam Abduladeem Kamil, Simulated performance of TCP, SCTP, DCCP and UDP protocols over 4G network, Procedia Computer Science, Volume 111, 2017