UNIVERSITY
OF OSLO

Master's thesis

TCP PEP

Extension of a TCP Performance Enhancing Proxy to Support Non-interactive Applications

Joe Bayer

Informatics: Programming and System Architecture
60 ECTS study points

Department of Informatics
Faculty of Mathematics and Natural Sciences

Spring 2023

Joe Bayer

TCP PEP

Extension of a TCP Performance
Enhancing Proxy to Support
Non-interactive Applications

Supervisors:
Michael Welzl
Kristjon Ciko

Contents

1 Intro **2**

2 Background **3**

2.1 Future of wireless communication. 3

2.1.1 5G 3

2.1.2 mmWave 3

2.1.3 Buffer bloat. (Buffering) 3

2.1.4 Non-Interactive Applications 4

2.2 TCP/IP 4

2.2.1 TCP Fast Open 4

2.2.2 0 RTT 4

2.2.3 Congestion control 4

2.3 PEPs 4

2.4 Kernel Modules 5

2.4.1 System Calls 5

3 Implementation | Design **6**

4 Evaluation **7**

5 Conclusion **8**

Chapter 1

Intro

Chapter 2

Background

2.1 Future of wireless communication.

2.1.1 5G

The future of wireless communication has seen a lot of improvements such as... ... highly increased bandwidth ... using Millimeter frequency bands ... but at the cost of Highly fluctuating bandwidth with wireless networks, especially with mmWave.

2.1.2 mmWave

s What is mmWave? Why? How? A big problem with mmWave communication is Line of sight blocking (signal path blocking). This is were ... Even the human body can create enough blockage to drastically reduce the bandwidth.

[Figure from mmwave paper showing bandwith fluctuations.]

”To achieve high throughput as well as low latency, these wireless networks will rely heavily on millimeter wave frequency bands (30-300 GHz), due to the large amounts of spectrum available on those bands.” (qoute mmwave paper)

”Applications that require extremely low latency are expected to be a major driver of 5G and WLAN networks that include millimeter wave (mmWave) links.” (qoute mmwave paper)

”Verizon’s mmWave network deployed in Minneapolis and Chicago reported a high handover frequency due to frequent disruptions in mmWave connectivity”(cite A First Look at Commercial 5G Performance on Smartphones)

2.1.3 Buffer bloat. (Buffering)

Large buffers, create buffer bloat, which is detrimental for latency sensitive applications. Preferred to drop packets and keep buffers small to avoid buffering time sensitive packets such as SYN packets.

Most focus has been on (helping? Supporting?) latency sensitive applications like virtual reality or remote surgery to name a few. This thesis will explore non-interactive applications where latency is not that critical and more buffering is acceptable and most likely desirable.

2.1.4 Non-Interactive Applications

Non-Interactive applications such as Web traffic, File transfers and Videos? can benefit from larger buffering, especially with fluctuating bandwidths. Being able to have packets buffered for when the bandwidth is high will decrease delay times. (need citation or prove it myself?)

2.2 TCP/IP

Interactive traffic uses TCP? source End to end argument. TCP handshake, reduce RTTs but using TCP Fast Open.
End to End congestion controller not very suited for highly fluctuating bandwidth.(cite David Hayes?)

2.2.1 TCP Fast Open

Short flows terminating in a few round-trips. Meaning the "bottleneck" is the required initial TCP handshake.
TCP Fast Open allows data being exchanged during the handshake.

2.2.2 0 RTT

Allow "syn fowarding" with TCP Fast Open creating a 0RTT increase when connecting through a proxy. 0RTT Transport Converter [1].

2.2.3 Congestion control

Congestion controller domains (different congestion controllers.) [2]. Wireless versus Wired networks, different congestion controllers needs. But domain split congestion control is not able to adapt fast enough to large changes in bandwidth. (cite David?)

2.3 PEPs

More logic inside the networks. Domain splitting and 0RTT. cite Kristjon Ciko?

2.4 Kernel Modules

LKM (Loadable Kernel Modules), "program" running inside the Linux kernel. Userspace vs Kernel.

2.4.1 System Calls

Reduce over head from userspace -> kernel system calls.

Chapter 3

Implementation | Design

Chapter 4

Evaluation

Chapter 5

Conclusion

Bibliography

[1] Olivier Bonaventure, Mohamed Boucadair, Sri Gundavelli, SungHoon Seo, and Benjamin Hesmans. 0-RTT TCP Convert Protocol. RFC 8803, July 2020.

[2] M. Welzl and W. Eddy. Congestion control in the rfc series. RFC 5783, RFC Editor, February 2010. `https://www.rfc-editor.org/info/rfc5783`.