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

Chapter 1

Intro

Chapter 2

Background

2.1 Future of wireless communication.

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

2.1.1 5G Millimetre Wave

The emergence of 5G Millimeter wave communications has opened the doors for low latency networks with multiple gigabit bandwidth. This is achieved by using higher millimetre wave (mmWave) frequencies in the range of 30GHz to 300GHz, which as a lot of benefits. [1] A wider spectrum of frequencies to choose from and higher data transfer are just some of the many benefits mmWave provides. But along side the benefits, mmWave has also introduced a lot of new challenges.

A big problem with millimetre wave communication is signal path blocking also called "Line of sight blocking" [5]. 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)

2.1.2 Buffer bloat. (Buffering)

The buffer bloat problem occurs when the systems between the endpoints buffer so many packets that the latency drastically increases and the reliability of the network as a whole goes down.https://lwn.net/Articles/507065/ The (AD HOC) crude but simple solution was to just decrease all buffer sizes. Although this works in most cases, its far from a optimal solution. The increased latency is detrimental for interactive (latency sensitive) applications. Generally its 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. By splitting traffic into interactive and non-interactive we can improve the performance of both. By having very small buffers for interactive applications we avoid bufferbloat problems, while utilizing the benefits of big buffers for non-interactive applications.

2.1.3 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 (Move to top?

(Get inspired by intro of tcp transport converter) Perhaps the most well known internet transport protocol is the Transmission Control Protocol (TCP). It provides reliable and in order delivery of packets using acknowledgments (ACKs) and re-transmissions. [4] Interactive traffic uses TCP? source End to end argument. Bad? TCP handshake, reduce RTTs but using TCP Fast Open. Mostly short flows (cite)
End to End congestion controller not very suited for highly fluctuating bandwidth.(cite David Hayes?)

2.2.1 0 RTT (3 Way handshake)

For TCP to establish a connection it uses a three-way handshake. First it sends a synchronization (SYN) packet to the desired endpoint. The endpoint answers with a acknowledgement and a synchronization packet of its own (SYN/ACK). And finally the client responds with a acknowledgment (ACK). At this point both endpoints have confirmed that they are ready for further communication. For any connection to be established this handshake has to be done. If we introduce a proxy we will need two handshakes, one for the initial connection and one for the connection from he proxy to the endpoint.

2.2.2 TCP Fast Open

A TCP connection can be configured with optional header extensions called TCP Options. 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.

Allow "syn fowarding" with TCP Fast Open creating a 0RTT increase when connecting through a proxy. 0RTT Transport Converter [2].

2.2.3 Congestion control

What is congestion controll? SHORT INTRO Different algorithms (QUBIC, RENO, VEGAS, BBR). Congestion controller domains (different congestion controllers.) [6]. 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

A performance enhancing proxy (PEP) is a (connection splitting) proxy designed to increase performance of applications using it. Already in use! (satelites, cite). Inherently increases performance? (cite) More logic inside the networks. Domain splitting and 0RTT.

2.3.1 Transparent vs Non-Transparent

A big discussion regarding PEPs has been if they should be transparent or non-transparent. Transparent PEPs are not visible to the applications that use it. They silently split the connections and spoof the IP-address of both the client and server [3].. (SIDE EFFECTS). Non-Transparent PEPs on the other hand are explicitly chosen by either the client or the server, and the sender is aware of the proxy splitting the original connection. This approach can be seen as more ethical and "correct" (FIND BETTER WORDING), but modifications at the sender side become necessary to utilize the PEP.

2.4 Kernel Modules

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

2.4.1 System Calls

Reduce over head from userspace -> kernel system calls.

Chapter 3

Implementation | Design

Table of design decisions based on different PEP implementations compared to ours. 0RTT, Transparent, Using TLV, Special ACKS, connection splitting.

Country List				
Implementation	0Rtt	Connection Splitting	Special ACKs	Transparent
Afghanistan	AF	AFG	004	x
Aland Islands	AX	ALA	248	x
Albania	AL	ALB	008	x
Algeria	DZ	DZA	012	x
American Samoa	AS	ASM	016	x
Andorra	AD	AND	020	x
Angola	AO	AGO	024	x

Chapter 4

Evaluation

Chapter 5

Conclusion

Bibliography

[1] S. K. Agrawal and Kapil Sharma. 5g millimeter wave (mmwave) communications. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, page 3630–3634, Mar 2016.

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

[3] Kristjon Ciko, Michael Welzl, and Peyman Teymoori. Pep-dna: A performance enhancing proxy for deploying network architectures. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, pages 1–6, 2021.

[4] Wesley Eddy. Transmission control protocol (tcp). Request for Comments RFC 9293, Internet Engineering Task Force, Aug 2022.

[5] Cristian García Ruiz, Antonio Pascual-Iserte, and Olga Muñoz. Analysis of blocking in mmwave cellular systems: Application to relay positioning. *IEEE Transactions on Communications*, 69(2):1329–1342, Feb 2021.

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