UNIVERSITY OF STAVANGER

BACHELOR THESIS

DATBAC

Make the Internet Faster!
Improving Alternative Backoff with ECN in Linux

Students
Dan Erik RAMSNES
Erlend Moen AL-KASIM

Supervisor
Naeem KHADEMI

February 13, 2020

Abstract

Lorem ipsum sit amet, consectetur adipiscing elit. Sed mollis dolor risus, a pulvinar quam pretium et. Suspendisse ut dolor arcu. Nulla facilisi. Nullam finibus vestibulum nulla, ac sollicitudin neque pretium vitae. Donec in est pretium, elementum velit et, pulvinar enim. Vestibulum consectetur et lectus ut fringilla. Donec malesuada, ante quis ultricies feugiat, leo tortor egestas nunc, non sagittis nisl lorem ac felis. Donec venenatis eget tortor et aliquam. Integer a sapien ultricies, dapibus erat sit amet, luctus ex.

Donec vitae metus pretium, tempus sapien eget, maximus quam. Nam dictum aliquam mi, ut fringilla nunc vehicula a. Morbi vitae dictum eros. Phasellus non urna felis. Etiam eu lectus justo. Etiam et ex ultrices, elementum erat laoreet, hendrerit lorem. Vivamus imperdiet consectetur dictum. Sed ac orci placerat quam rutrum pellentesque. Proin sollicitudin diam et erat feugiat feugiat. Quisque tempus velit viverra sem aliquet feugiat vitae et augue. Donec at odio viverra, posuere tortor eu, volutpat quam. Proin id rutrum metus, id mollis ipsum.

Sed pulvinar tristique nibh eu convallis. Integer luctus pretium massa sit amet placerat. Donec tincidunt consectetur efficitur. Nam tincidunt libero ut nisi lacinia, rhoncus cursus elit lobortis. Interdum et malesuada fames ac ante ipsum primis in faucibus. Duis vel semper lectus. Donec et ullamcorper turpis.

Contents

Chapter 1

Introduction

This chapter aims at giving an introduction and overview of the thesis. It starts with a brief explanation of why Internet today still feels slow despite major advances in technology, followed up by establishing the goals and research questions for the entire thesis. To address the research questions, a small look into the research methodology is presented. In the final section, an outline of the thesis structure is discussed.

Donec vitae metus pretium [1], tempus sapien eget, maximus . Nam dictum aliquam mi, ut fringilla nunc vehicula a. Morbi vitae dictum eros.

1.1 Motivation

Maecenas pulvinar quis quam eu convallis. Fusce vulputate sodales suscipit. Nullam porttitor hendrerit lacinia. Cras tincidunt ultrices lobortis. Sed ac sem efficitur, faucibus nunc vehicula, elementum arcu. Aenean pellentesque augue ut massa ullamcorper congue. Phasellus varius at erat at lobortis. Ut nunc metus, consequat nec blandit in, laoreet nec ligula. Integer interdum, massa eu accumsan eleifend, nunc nunc ultrices augue, at commodo risus ligula varius ex. Praesent blandit risus sit amet tellus semper, a suscipit libero malesuada. Mauris nec lacus ipsum. Nullam at ipsum rhoncus, mattis nibh vitae, placerat odio. Sed nec leo id dui vulputate accumsan. Pellentesque placerat congue arcu id pharetra. Maecenas imperdiet ex sed vehicula euismod.

1.2 Goals and Research Questions

Aliquam quis imperdiet orci, nec vestibulum risus. Curabitur vitae euismod mauris, ut iaculis erat. Ut interdum ex ac elit ultrices, at mattis magna ultricies. Duis ac suscipit nibh. Mauris sagittis consequat elit eget euismod. Mauris tincidunt dui ex, sit amet vulputate sem pretium non. Quisque at nulla lobortis, facilisis ligula eu, efficitur ante. Praesent bibendum dolor purus, non placerat felis pellentesque eu.

1.3 Research Methodology

1.4 Contributions

1.5 Thesis Structure

Chapter 2

Literature Review

This chapter presents the background theory for which this thesis is based upon.

2.1 Network Delay and Bufferbloat

A common cause of latency in packet switched networks is bufferbloat. It occurs when a router, with a large buffer gets congested. the tcp Congestion Control (CC) will fill up the entire buffer, before it starts backing off. Packets become queued for a long period of time, untill the buffer is drained, CC resets and TCP connection ramps back up to speed to fill the buffer again.

This causes high and variable latency, in addition to "blocking" the bottleneck for other flows when the buffer is full and packets are droped. Several technical solutions exists, that try to solve the problem of bufferbloat and we will outline some of them in this section.

2.2 Transmission Control Protocol

Transmission Control Protocol (TCP) is the one of the main protocols for transmitting data on the internet. It is a connection based, reliable protocoll and is used by for instance World Wide Web (WWW), email, File Transfer Protocoll (FTP) and streaming media. TCP requires that the sender and reciever establishes a connection through a three-way handshake before transmission starts. All segments sent have a sequence number, and the reciver sends an Acknowledgement (ACK) for every segment it recieves. This, in addition to retransmission and error-checking ensures reliable transfer, but also lengthens latency.

2.2.1 Congestion Control

To help reduce congestion on the links TCP maintains a Congestion Window (CWND), which limits the total number of unacknowledged packets that it can send at a time. This is done in multiple fases.

In the slow start fase, right after a connection is established. the congestion window starts as a small multiple of Maximum Segment Size (MSS) and is effectivley doubled for every Round Trip Time (RTT). When it reaches the slow-start threshold(ssthresh), CWND is reduced by half and a new fase starts, congestion avoidance. In this fase CWND is increased linearly by one MSS every RTT. If loss occurs, it could mean there is congestion, and steps will be taken to reduce load on the network. The steps depend on what exact congestion avoidance algorithm is used.

2.2.2 Active Queue Management

2.3 Explicit Congestion Notification

Explicit Congestion Notification (ECN) is an extension to TCP that allows routers to notify end points on impending congestion without dropping packets.

2.3.1 Legacy ECN

In legacy ECN, the router notifies end hosts of congestion by setting a Congestion Encountered (CE) flag in the IP header on ECN enabled packets when experiencing congestion. The reciever of the packet then reflects this back to the sender by setting an ECN-Echo (ECE) in the TCP header. It keeps doing this until the sender responds back with a segment with Congestion Window Reduced (CWR) set, indicating that the sender has backed off.

2.4 Alternative Backoff with ECN

Chapter 3

Methodology

3.1 Network Topology

3.1.1 Raspberry Pi 4 Cluster

3.2 TCP Experimentation with TEACUP

3.2.1 Exposing TCP State with web10g

3.3 Achieving Low Latency with ABE

3.4 Improving ABE by Adapting Its Reduction Factor β

Chapter 4

Results

Chapter 5

Conclusion

Appendix A

The PI4-Cluster Testbed

A.1 Setting Up Dual Boot

First install Ubuntu. When asked for partitioning the disk, choose manual, select the disk and confirm creating a new empty partition with yes. Select the newly created empty partition followed by create a new partition and set a size for it. The type should be of primary, location at beginning and mounting point root. Finish off with done setting up the partition followed by finish partitioning and write changes to disk.

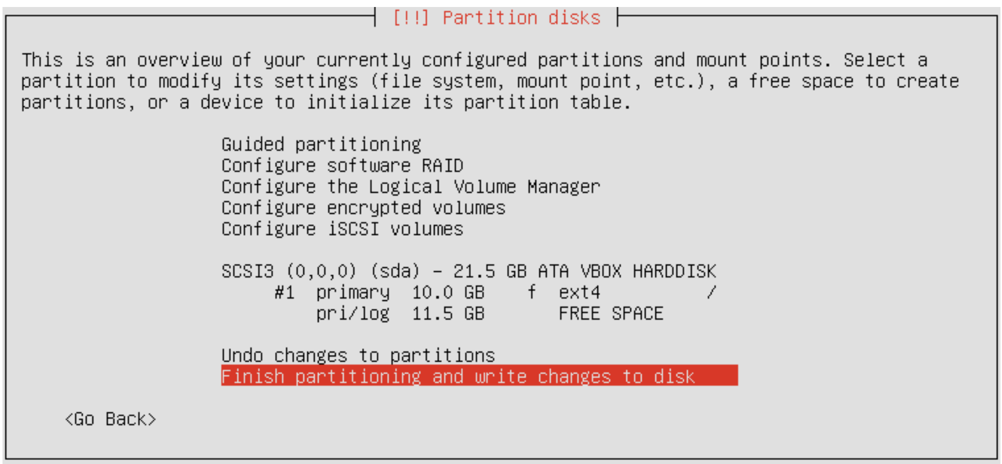

Figure A.1: The partition editor for Ubuntu.

Next, install FreeBSD. When asked for partitioning the disk, choose **auto** (UFS) followed by partition. Set a size, hit ok and finish.

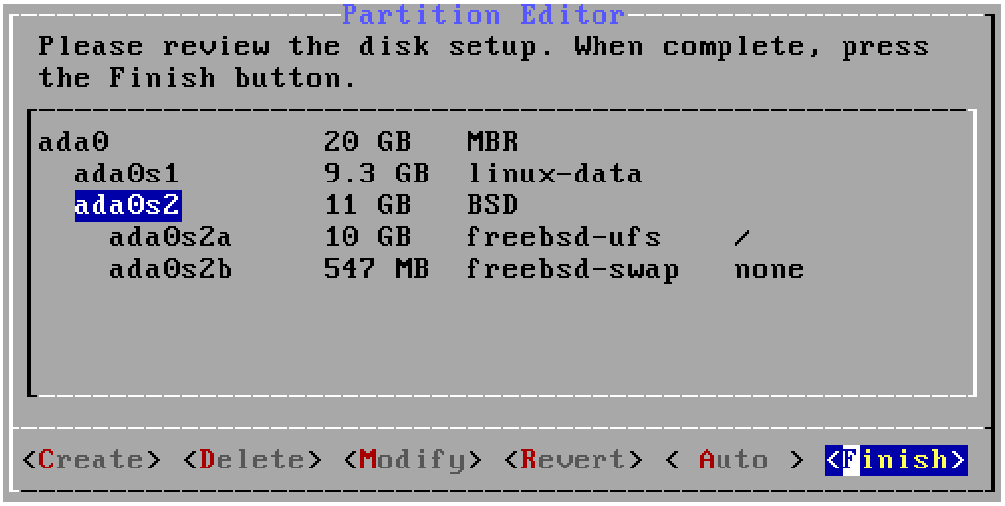

Figure A.2: The partition editor for FreeBSD.

After installing both systems, only Ubuntu is presented in the GRand Unified Bootloader (GRUB). To add FreeBSD as an option, run `sudo nano /etc/grub.d/40_custom` in Ubuntu, and add the following entry:

```
menuentry "FreeBSD" {
    insmod ufs2
    set root=(hd0,2)
    kfreebsd /boot/loader
}
```

Then update GRUB with `sudo update−grub`. The FreeBSD option should now be available when rebooting. If the bootloader won't display, hold the RIGHT SHIFT key upon booting.

To enable a one-time reboot into FreeBSD from Ubuntu, run the command `grub−editenv /boot/grub/grubenv set next_entry="FreeBSD"` and reboot with `sudo reboot`.

A.2 Compiling Mainline Kernel 5.5 for Raspberry Pi 4

A.3 Patching web10g on Mainline Kernel 5.5

Terms

Acknowledgement (ACK) A signal that is passed between communicating processes, computers, or devices to signify acknowledgement, or receipt of message, as part of a communications protocol.

Congestion Control (CC) The process of managing the sender's packet rate to not overwhelm the network.

Congestion Window (CWND) A TCP state variable that limits the amount of data the TCP can send into the network before receiving an ACK.

Explicit Congestion Notification (ECN) An extension to IP and TCP that allows end-to-end notification of network congestion without dropping packets.

GRand Unified Bootloader (GRUB) A Multiboot boot loader. It was derived from GRUB, the GRand Unified Bootloader, which was originally designed and implemented by Erich Stefan Boleyn.

Maximum Segment Size (MSS) The largest specified amount of data (in bytes) that a communications device can receive in a single TCP segment.

Round Trip Time (RTT) The time it takes for a signal to be sent plus the time for an ACK of that signal to be received.

Transmission Control Protocol (TCP) One of the main communication protocols of the Internet that defines how to establish and maintain a network conversation through which applications can exchange data.

References

[1] Naeem Khademi et al. "Alternative Backoff: Achieving Low Latency and High Throughput with ECN and AQM". In: (2017), p. 9.