



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Jaakko Samuel Rotonen

VERKKOLIIKENTEEN RUUHKATUMISEN HALLINNAN RATKAISUT JA MENETELMÄT

Tekniikka
2023

TIIVISTELMÄ

Tekijä	Jaakko Samuel Rotonen
Opinnäytetyön nimi	Verkkoliikenteen ruuhkautumisen hallinnan ratkaisut ja menetelmät
Vuosi	2023
Kieli	suomi
Sivumäärä	44
Ohjaaja	Antti Virtanen

Kun lähiverkossa liikkuu paljon dataa, osa siitä joudutaan puskuroimaan reitittimen toimesta ennen kuin se voidaan lähettää eteenpäin. Verkossa liikkuvan datan liiallinen puskurointi aiheuttaa lähiverkossa viivettä ja verkkopakettien tippumista, jota kutsutaan yleisesti nimikkeellä Bufferbloat.

Tässä opinnäytetyössä mitattiin liiallisesta puskuroinnista aiheutuvaa viivettä kuormitetussa lähiverkkoyhteydessä, sekä yhteyteen muodostuneen viiveen vähentämistä reititinlaitteeseen käyttöönotettavilla ohjelmistopohjaisilla ratkaisuilla ja sen vaikutusta verkon lataus- ja lähetysnopeuksiin kuluttajatasoisella reititinlaitteella. Työssä verrattiin kahta ohjelmistopohjaista ratkaisua toisiinsa kahta eri verkkonopeutta tarjoavalla verkkoyhteydellä.

Saatujen tuloksien avulla arvioitiin liialliseen puskuroimiseen kehitettyjen ohjelmistoratkaisujen tehokkuutta ja käytiin läpi miten kyseiset ratkaisut toimivat. Työssä lisäksi kerrottiin mistä käytetyt ohjelmistoratkaisut löytyvät, miten niitä käytetään ja muodostettiin johtopäätös kyseisten ratkaisujen käyttöönoton kannattavuudesta kuluttajan näkökulmasta.

Tulokset osoittivat, että reitittimellä käyttöönotettavat ohjelmistopohjaiset ratkaisut auttoivat hillitsemään kuormitettuun verkkoyhteyteen muodostuvaa viivettä 150/50 Mbit/s tasoisessa verkkoyhteydessä. Nopeammassa verkkoyhteydessä kuitenkin todettiin testattujen ratkaisujen olevan riittämättömiä reititinlaitteen ollessa kykenemätön hyödyntämään ohjelmistoratkaisujen algoritmeja riittämättömän prosessointikyvyn takia.

ABSTRACT

Author	Jaakko Samuel Rotonen
Title	Methods and Measures to Control Network Traffic Congestion
Year	2023
Language	Finnish
Pages	44
Name of Supervisor	Antti Virtanen

When a lot of data is being transmitted through a local area network, some of it must be buffered by the network router before it can be forwarded to its destination. Excess amount of buffering by the router causes delay on the network in addition to causing network packets to be dropped. The latency and package dropping from excess buffering is commonly referred to as Bufferbloat.

This thesis was carried out with the purpose of measuring the latency caused by Bufferbloat on a strained local area network connection, to test out the effectiveness of the features designed to mitigate the issue and how these features affect the networks download and upload speeds. Two different solutions were tested and compared to each other using two different fixed broadband connection tiers.

By analyzing the results of the tests, conclusions were drawn on the effectiveness of these measures to manage Bufferbloat and how they affect network transfer speeds. The thesis also describes how these measures against Bufferbloat work, how to acquire and use these features and a conclusion was drawn whether these features should be used, from a consumer's perspective.

The results suggested that the software based solutions helped mitigate the latency caused by Bufferbloat on a strained broadband connection with 150/50 Mbps speeds. On a faster network connection, the solutions did not result in a favorable result, when the router was unable to utilize the algorithms provided by the software based solutions due to the lack of processing capabilities.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVALUETTELO

LYHENTEET

1	JOHDANTO.....	8
2	TAUSTA, TARKOITUS JA TAVOITTEET	9
3	TEORIATAUSTA	11
	3.1 Bufferbloat	11
	3.2 NAT Acceleration	12
	3.3 Netgear QoS.....	14
	3.4 SQM.....	15
	3.5 CoDel.....	17
	3.6 FQ-CoDel	18
	3.7 CAKE	20
4	OPENWRT-LAITEOHJELMISTON ASENNUS	24
5	LIIALLISEN PUSKUROINNIN TESTAAMINEN	27
	5.1 Laitteet	27
	5.2 Ohjelmisto.....	27
	5.3 Kytkennät	27
	5.4 iPerf3	28
	5.5 Testausmenetelmä	30
6	TULOKSET	34
7	TULOSTEN ANALYYSI	38
8	JOHTOPÄÄTÖKSET	41
	LÄHTEET	42

KUVALUETTELO

Kuva 1. OpenWrt Routing/NAT Offloading -asetussivu.....	14
Kuva 2. Netgear QoS -asetussivu	15
Kuva 3. SQM Queue Discipline -välilehti	17
Kuva 5. OpenWrt-laiteohjelmiston asennus Netgear-laiteohjelmistopäivityksen avulla	24
Kuva 6. Reitittimeen kytkettyjen koneiden staattiset IP-osoitteet ja leasing ajat OpenWrt-laiteohjelmiston DHCP/DNS-asetussivulla.....	25
Kuva 7. SQM-ohjelmistopakettien asennusikkuna tietoineen	26
Kuva 8. Laitekytkentöjen topologia	28
Kuva 9. iPerf3-palvelin käynnistyskomento.....	29
Kuva 10. iPerf3-asiakasohjelma	30
Kuva 11. Työssä käytettyjen Elisan laajakaistaliittymien yhteysnopeudet esitettynä Elisan verkkosivuilla.	31
Kuva 12. WaveForm Bufferbloat -testin tulos.	33
Kuva 13. Verkon siirtonopeuksia kolmella eri mittauksella, joiden keskiarvosta muodostuu taulukon esittämät tulokset	35
Kuva 14. Verkossa mitatut viiveet eri testiasetuksin kolmella eri mittauksella, joiden keskiarvosta muodostuu taulukossa esiintyvät tulokset	37
Kuva 15. Verkon kokonaissiirtonopeus suhteessa mitattuun viiveeseen	39

LYHENTEET

ACK	ACKnowledgement, kuittaus onnistuneesta verkkopaketin siirrosta TCP-IP-protokollan tiedonsiirrossa
AQM	Active Queue Management, aktiivinen jononhallinta
ARPANET	Advanced Research Projects Agency Network, Yhdysvaltalainen pakettikytkennäinen tietokoneverkko
CAKE	Common Applications Kept Enhanced, Bufferbloat ilmiötä vastaan kehitetty jononhallintajärjestelmä
CoDel	Controlled Delay, verkkopakettien jononhallinta-algoritmi
DHCP	Dynamic Host Configuration Protocol, verkkoprotokolla, jonka avulla lähiverkkoon kytketyt laitteet saavat verkkoasetukset
FIFO	First In First Out, jonotustapa, jossa jonoon saapuvat paketit käsitellään tulojärjestyksessä
FQ-CoDel	Flow Queue CoDel, verkkopakettien vuorontamisesta ja aktiivisesta jononhallinnoimisesta muodostuva algoritmi
NAT	Network Address Translation, osoitteenmuunnostekniikka, jonka avulla yhden IP-osoitteen taakse saadaan monta osoitetta
QoS	Quality of Service, tietoliikenteen priorisointijärjestelmä

SQM	Smart Queue Management, Bufferbloat ongelmaa vastaan kehitetty verkkopakettien jononhallintajärjestelmä
TCP	Transmission Control Protocol, tietoliikenneprotokolla, joka varmistaa verkkopakettien saapumisen määrän- päähän

1 JOHDANTO

Nykyaikaiset tietoliikenneyhteydet toimivat pakettikytkennällä, joka juontaa juurensa jo ARPANETin aikoihin. Pakettikytkennässä data jaetaan paketeiksi tiedonsiirtoa varten ja paketit välittyvät verkossa lähettäjältä vastaanottajalle otsikkokentässä olevien osoitetietojen perusteella. Pakettikytkennäisessä tiedonsiirrossa tiedonsiirron määrän kasvaessa liian suureksi, osa dataa siirtävistä verkkopaketeista joudutaan puskuroimaan verkkolaitteiden toimesta odottamaan vuoroaan. Pakettikytkennäinen tiedonsiirto tarvitsee puskurointia toimiakseen luotettavasti verkkoyhteyden tiedonsiirtokapasiteetin muuttuessa, mutta liiallisen puskuroinnin on havaittu aiheuttavan viivettä tietoliikenteeseen. Ensimmäiset havainnot puskuroinnin aiheuttamasta verkon latenssista on tehty jo vuonna 1985 John Naglen toimesta.¹ Liiallista datan puskurointia kutsutaan nimikkeellä Bufferbloat, jonka on havaittu aiheuttavan ongelmia erityisesti kotikäyttöön suunnatuissa kuluttajaverkkolaitteissa.² Tietoliikenneverkkojen liikenteen määrän kasvaessa vuosi vuodelta, herää tarve tarkastella lähiverkkoyhteyksissä kehittyvää viivettä ja sitä vastaan kehiteltyjä ratkaisuja.

Tässä opinnäytetyössä mitataan liiallisesta puskuroinnista aiheutuvaa viivettä lähiverkkoyhteydessä, tutustutaan ratkaisuihin puskuroinnin hallitsemiseen ja mitataan näiden ratkaisujen tehokkuus suhteessa verkkoon, jossa näitä ratkaisuja ei hyödynnetä. Työssä lisäksi tutustutaan liiallista puskurointia vastaan kehitettyjen ratkaisujen toiminnallisuuteen, kerrotaan näiden ohjelmistoratkaisujen käyttöönotosta, mitataan niiden vaikutusta verkkoyhteyden lataus- ja lähetysnopeuksiin ja arvioidaan niiden hyödyllisyys kuluttajan näkökulmasta.

¹ Nagle, J. 1985. On Packet Switches With Infinite Storage

² Allman, M. 2012. Comments on bufferbloat

2 TAUSTA, TARKOITUS JA TAVOITTEET

Työn tausta perustuu tietoverkkoyhteyksiä vaivanneeseen ongelmaan, joka kulkee nimikkeellä Bufferbloat. Bufferbloat ei ilmiönä ole mikään uusi ja sitä ehkäisemään on kehitetty monenlaisia ratkaisuja. Bufferbloat ilmiötä vastaan kehitettyjen ratkaisujen käyttöön ottamisesta, kokeilusta ja vertailusta keskenään ei kuitenkaan olla juurikaan raportoitu kuluttajatasen laitteilla. Tämän työn tarkoituksena on kokeilla kahta yleistä ja helposti käyttöön otettavaa ratkaisua sekä vertailla niiden tehokkuutta liiallisen puskuroimisen välttämiseksi, sekä miten ne vaikuttavat verkon siirtonopeuksiin. Mittauksia analysoimalla ja lähiverkkoyhteyteen liittyvien käsitteiden läpi käymisellä pyritään selventämään mistä Bufferbloat-ilmiössä on kyse, sekä käydään läpi miten liiallista puskurointia vastaan kehitetyt ratkaisut toimivat. Lisäksi työssä selvitetään, onko puskuroinnin hallinnointiin kehitettyjen ratkaisujen käyttöönotto kannattavaa.

Käytämme vuosi vuodelta enemmän dataa ja tästä syystä on hyvä odottaa liiallisen puskuroinnin olevan yleistävä ongelma lähiverkkoyhteyksissä.³ Monissa reititinlaitteissa on tarjolla ominaisuus reitittimen puskurin hallinnoimiselle, mutta tämä asetus on oletusasetuksilla lähes aina kytketty pois päältä. Tästä syystä tässä työssä käydään läpi myös liiallisen puskuroinnin välttämiseksi kehitettyjen ratkaisujen käyttöönotto, sekä kokeillaan vaihtoehtoja kuten OpenWrt-laiteohjelmistoa, koska kaikista reititinlaitteista ei löydy käyttöönotettavaa ominaisuutta hallinnoimaan reitittimen puskurin muodostuvia verkkopakettijonoja.

Työssä käytettiin 9 vuotta vanhaa Netgear R6220 -reititinlaitetta, joka vastaa hyvin sen ajan keskivertoreititinlaitetta (v. 2014). Kyseinen reititinlaite tai vastaavan ta-

³ OpenVault. 2022. Broadband Insights Report (OVBI) 1Q22

soinen reititinlaite on todennäköisesti vielä valtaosalla sen ostajista käytössä tänäkin päivänä. Kyseiseltä Netgear -reitittimeltä, edellä mainitun lisäksi, löytyy QoS-ominaisuus oletuslaiteohjelmistosta, jonka tehokkuutta pystyy vertaamaan OpenWrt-laiteohjelmiston vastaavaan ratkaisuun. Reitittimen vanhentuneitten komponenttien takia, tällä reitittimellä voi havaita verkkoyhteydessä ilmaantuvia Bufferbloat -ongelmia, joten se soveltuu testauksiin hyvin. Reititinlaitteille saatavissa laiteohjelmistoissa on kaksi muuta vaihtoehtoa OpenWrt-ohjelmiston lisäksi, kuten DD-WRT ja Tomato, joista kumpikaan ei valitettavasti tarjonnut työssä käytettyin reititinlaitteeseen tukea. OpenWrt-laiteohjelmisto tarjoaa lisäksi helpomman käyttöliittymän näihin kahteen muuhun laiteohjelmistoon verrattuna, joten sen katsottiin olevan sopivin ratkaisu työn tavoitteiden näkökulmasta.

Tavoitteena työllä on kuvailla Bufferbloat-ilmiötä, liiallista puskurointia vastaan kehitettyjen ratkaisujen toiminnallisuutta, sekä arvioida niiden toiminnallisuutta mittaustulosten avulla. Mittaamalla reititinlaitteessa mukana tulevaa verkkoliikenteen hallintamenetelmän tehokkuutta modernimpaan laiteohjelmistoratkaisuun, jonka pystyy asentamaan melkein mihin tahansa reitittimeen, saadaan aikaan selkeä katsaus tarjottujen ratkaisujen tehokkuudesta hallita verkkoliikennettä, sekä annetaan edellytykset lukijalle kokeilla näitä ratkaisuja omalla laitteellaan halutessaan.

3 TEORIATAUSTA

3.1 Bufferbloat

Bufferbloat käsitteenä viittaa latenssiin, joka johtuu reitittimen liiallisesta verkkopakettien puskuroimisesta.⁴ Liiallinen puskurointi tapahtuu reitittimen toimiessa pullonkaulana siihen kytkettyjen laitteiden ja ulkoverkon välillä. Verkkoliikenteen saavuttaessa tason, jossa se ylittää reitittimen kyvyn käsitellä sen kautta kulkevia verkkopaketteja, reititin joutuu puskuroimaan osan verkkopaketeista jonoon odottamaan vuoroaan. Verkkopakettien puskuroiminen on pakettikytkennäiselle tiedonsiirrolle välttämätöntä varmistamaan verkkopakettien saapumisen määränpäihinsä verkkoliikenteen tiedonsiirtokapasiteettien muuttuessa tiedonsiirron aikana.⁵ Verkossa kulkevan liikennemäärän kasvaessa liian suureksi, verkkopakettien puskuroiminen on siis pakollista, mutta se aiheuttaa luonnostaan latenssia verkossa toimivaan tiedonsiirtoon ja tilan loppuminen reitittimen puskurista johdtaa verkkopakettien tippumiseen, kun ne eivät enää mahdu reitittimen täyteen puskuriiin.

Reitittimen liiallisen puskuroimisen vaikutuksia useimmiten havaitaan tietokonesovelluksissa, joiden toiminnallisuus on riippuvainen pienestä latenssista. Sovellukset, joissa hyvä latenssi on välttämättömyys, ovat muun muassa verkkopelit, verkkopuhelut sekä videopuhelut. Korkea verkkoviive näissä sovelluksissa havaitaan yleensä yhteyden pätkimisenä tai yhteyden kokonaan tippumisena. Liiallisen puskuroimisen vaikutukset eivät kuitenkaan vaikuta pelkästään pientä latenssia vaativiin sovelluksiin, vaan sen voi helposti huomata verkkoa selaillessakin uuden

⁴ Bufferbloat.net. 2022. More about Bufferbloat

⁵ Nichols, K. & Jacobson, V. 2012. Controlling Queue Delay

verkkosivun avautumiseen kuluvan ajan vaihdella jatkuvasti verkkoliikenteen latenssimuutosten takia.⁶

3.2 NAT Acceleration

Ennen Bufferbloat-ongelmaa vastaan kehitettyjen ratkaisujen läpikäyntiä on hyvä tietää mihin termeillä NAT Acceleration, Hardware Acceleration, Software ja Hardware Offloadingilla viitataan, sekä niiden vaikutuksesta reitittimen tiedonsiirtonopeuteen. Useissa reitittimissä on jonkinlainen NAT Acceleration tai toiselta nimeltään Hardware Acceleration -ominaisuus käytössä, helpottamaan reitittimen komponenttien työtaakkaa, mikä vuorostaan mahdollistaa suuremman tiedonsiirtonopeuden saavuttamisen reitittimillä. Hardware/NAT Acceleration -toiminto jättää osaa TCP/IP-tietoliikenneprotokollapinon määritelmistä käyttämättä vähentäen reitittimen komponenteilta vaadittua työmäärä sen käsitellessä ulkoverkkoon siirtyviä verkkopaketteja. Kyseinen kiihdytystoiminto on jaettu reitittimissä ohjelmisto- ja laitteistopohjaisiin ratkaisuihin, joihin ei tässä opinnäytetyössä syvemmin tutustuta, koska niiden toimintaa ei olla avattu tarkemmin Netgearin tai OpenWrt:n toimesta. Netgearilla Hardware/NAT Acceleration on automaattisesti päällä ja OpenWrt:n tapauksessa Hardware/NAT Acceleration on jaettu ohjelmistopohjaiseen (Software Offloading) sekä laitteistopohjaiseen (Hardware Offloading) valintaan OpenWrt-järjestelmän tarjoamilla reitittimen asetussivuilla, joten halutesaan reititintä voi käyttää ilman näitä kiihdytystoimintoja.⁷

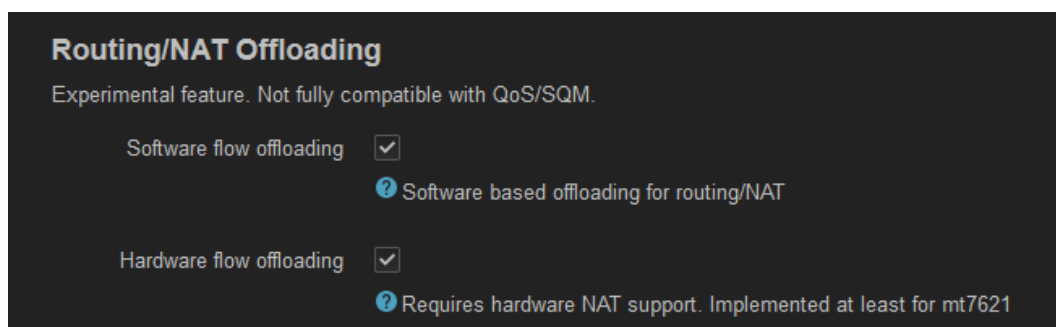
Se miten Hardware/NAT Acceleration liittyy liiallisen puskuroinnin estämiseen, on niiden yhteensopivuudessa QoS- ja SQM-toimintojen kanssa. QoS- ja SQM -ominaisuuksien toiminnallisuus perustuu reitittimen kautta kulkevien verkkopaket-

⁶ Gettys, J. 2011. Bufferbloat: Dark Buffers in the Internet

⁷ dd-wrt.com. 2022. Hardware

tien tarkasteluun ja hallinointiin, joka sulkee Hardware/NAT Acceleration -toiminnon yhteensopivuuden näiden kanssa pois. Netgearin laiteohjelmistolla Hardware/NAT Acceleration kytkeytyy pois päältä itsestään, kun QoS -ominaisuuden aktivoi ja OpenWrt -laiteohjelmiston asetussivuilla ilmoitetaan suoraan Offloading-toimintojen olevan vielä kokeellisia toimintoja ja ne eivät ole täysin yhteensopivia SQM -ominaisuuden kanssa. OpenWrt:n Routing/NAT Offloading -asetussivu esitetään alla kuvassa 1. QoS tai SQM päällä reititin ei siis kykene saavuttamaan maksimaalisia tiedonsiirtonopeuksia, koska se joutuu käsittelemään kaikki reitittimen kautta kulkevat verkkopaketit kasvattaen reitittimen komponenttien työmäärää ja verkkonopeudet jäivät Netgearin R6220 -reitittimellä noin puoleen SQM/QoS -ominaisuuksien ollessa päällä, kuten työssä esitetyistä tuloksista voi huomata. Netgear-reitittimen hitaammasta tiedonsiirtonopeudesta on erikseen raportoitu myös OpenWrt:n verkkosivuilla, jonka lisäksi tässä opinnäytetyössä testattavassa reitittimessä on suositeltu käyttämään reitittimen prosessorille kevyempää FQ-CoDel-algoritmia SQM-järjestelmässä modernimman CAKE-algoritmin sijaan.⁸

⁸ OpenWrt.org. 2021. NETGEAR R6220.



Kuva 1. OpenWrt Routing/NAT Offloading -asetussivu

3.3 Netgear QoS

Quality of Service -liikennehallinta perustuu määrittämällä verkkoon kytkettyjen laitteiden tai laitteilla toimivien ohjelmien verkkoliikenteen tärkeysjärjestys, jonka mukaan reititin siirtää verkkoliikennettä.⁹ QoS -ominaisuuden sisältävät reitittimet tarjoavat asetussivun, jonka avulla voit luokitella verkkoon kytkettyjen laitteiden tai laitteilla käytettävien ohjelmien prioriteetin. Reititin priorisoi korkeammalle merkittyjen laitteiden ja ohjelmien verkkoliikennettä jättäen matalammalle luokitetut odottamaan vuoroaan puskurissa määrättyyn jonoon. Korkeimmassa luokassa olevat laitteet ja ohjelmat saavuttavat aina paremman toiminnallisuuden verkossa, koska QoS-järjestelmä käsittelee alemman luokan verkkopaketit vasta, kun ylempien luokkien verkkoliikenne on käsitelty kokonaan. QoS-järjestelmän avulla käsitelty liikenne voikin olla varsin hidasta ja epäluotettavaa alempaan tärkeysluokkaan määritellyille laitteille ja ohjelmille. QoS-priorisointijärjestelmässä samassa luokassa olevat laitteet ja ohjelmat vuorostaan toimivat FIFO-menetelmällä. FIFO merkitsee sitä, että reitittimelle saapuvat saman tärkeysluokan verkkopaketit käsitellään saapumisjärjestyksessä. Alla esitettyssä kuvassa 2 näkyy Netgear-reitittimen QoS-asetussivu ja sen tapa luokitella kytketyt laitteet tärkeysjärjestykseen.¹⁰ Kuvassa näkyy, kuinka mittauksien tekoon käytetty DESKTOP-

⁹ kb.netgear.com. 2022. How QoS improves performance

¹⁰ Jha, S. & Mahdub, H. 2002. Engineering Internet QoS

P4WKR2U-tietokone on asetettuna korkeimpaan prioriteettiiluokkaan, ja verkkoliikennettä generoivat koneet ovat asetettuna alimpaan. Mittausten aikana ei todennettu mitään eroa sillä, laittoiko korkeammalla prioriteettiiluokalla olevan laitteen Highest, High vai Normal -luokkaan. Laitteiden verkkoliikenteen priorisointi oli samaa tasoa luokasta huolimatta, ainoa tuloksiin vaikuttava seikka oli se, että laite oli korkeammassa luokassa kuin muut.

QoS Policy	Priority	Device Name	MAC Address
Pti_MAC_D4D4C6	Low	LFP3M0TK0	88AA C2D4D4C6
Pti_MAC_46C8A8	Low	PK3_EC3_P9-1	480F CF48C8A8
Pti_MAC_30B284	Low	PK3_EC3_P1-2	480F CF30B284
Pti_MAC_530773	Highest	DESKTOP-P4WKR2U	D45D64530773

QoS Policy for:

MAC Address:

Device Name:

Priority: Normal

Buttons: + Add, Reset

QoS rules:

Queue 1: Highest	Queue 2: High	Queue 3: Normal	Queue 4: Low
Pti_MAC_530773			Pti_MAC_D4D4C6 Pti_MAC_46C8A8 Pti_MAC_30B284

Buttons: Edit, Delete, Delete All

Kuva 2. Netgear QoS -asetussivu

3.4 SQM

Smart Queue Management on OpenWrt-ohjelmistojärjestelmän tarjoama ja kehittämä ratkaisu reitittimen liiallisen puskuroinnin hallintaan, jonka voi asentaa ohjelmistopakettina OpenWrt-järjestelmän asetussivuilla. SQM:n toiminnallisuus perustuu verkkopaketeista muodostuvien datavirtojen järjestämiseen erillisiin jonoihin vuorontajalla ja verkkopaketeista muodostuvan jonon hallintaan vuorontajan käyttämällä algoritmilla (Queuing discipline). Datavirta- ja jononhallintayhdis-

telmiä on kaksi valittavana, Cake (Common Applications Kept Enhanced) ja FQ-CoDel (Flow Queue CoDel), joista testattiin molempia tässä opinnäytetyössä. Edellä mainitun asetuksen lisäksi käyttäjä valitsee Queue setup -skriptin, joka lisää verkkopakettien käsittelyyn prioriteettiluokituksen ja joillain skriptivalinnoilla tuodaan tähän luokitteluun vielä mukaan verkon kaistanhallinta. Skriptivaihtoehtoja oli tarjolla viisi kappaletta ja kyseisistä vaihtoehtoista tässä työssä käytettiin `piece_of_cake.qos` ja `simplest.qos` -skriptejä. Skripteistä `piece_of_cake.qos` oli OpenWrt-sivuston suosittelu sekä modernin ja `simplest.qos` oli vuorostaan kevyin reititinlaitteen prosessorille. Skriptin keveyden merkitystä käydään läpi myöhemmässä luvussa.¹¹

SQM-jononhallintajärjestelmän asetussivuilla määriteltiin ensin mille laitteen verkkoportille tai verkkoportille puskuroinnin hallinnointi asetetaan käyttöön ja ulkoverkon lataus- ja lähetysnopeus. Verkon lataus- ja lähetysnopeuden on tarkoitus rajata OpenWrt-verkkosivuston ohjeistuksen mukaisesti 85–95 % verkon maksimi siirtonopeudesta.¹² Siirtonopeuksien rajaamisen tarkoituksena on antaa reitittimelle tilaa reagoida verkkoarjoajan verkkosiirtonopeuksien muutoksille siirtäen verkon pullonkaulan reitittimen hallinnoitavaksi.¹¹ Queue-Discipline -välilehdellä valitaan datavirta ja jononhallintamenetelmät ja Link Layer Adaptation -välilehden kautta annetaan SQM-järjestelmän kaistanleveydenhallinnalle tieto verkossa kulkevien pakettien koosta, jotta se voi laskea kuinka kauan pakettien siirtäminen kestää. Queue-Discipline-asetussivulla Per Packet Overhead-arvo jätettiin tyhjäksi OpenWrt-sivuston suositusten mukaisesti, koska testit suoritettiin kuituverkolla.¹¹ Kuvassa 3 esitetään vuorontajan ja luokittelu skriptin valintasivu SQM-asetussivuilla.

¹¹ Openwrt.org. 2022. SQM Details

¹² Openwrt.org. 2022. SQM (Smart Queue Management)

Smart Queue Management

With SQM you can enable traffic shaping, better mixing (Fair Queueing), active queue length management (AQM) and prioritisation on one network interface.

Queues Delete

Basic Settings **Queue Discipline** Link Layer Adaptation

Queueing disciplines useable on this system. After installing a new qdisc, you need to restart the router to see updates!

Queue setup script cake

Queue setup script piece_of_cake.qos

piece_of_cake.qos:
This just uses the cake qdisc as a replacement for both htb as shaper and fq_codel as leaf qdisc. It just does not come any simpler than this, in other words it truly is a "piece of cake". This script requires that cake is selected as qdisc, and forces its usage. See: <http://www.bufferbloat.net/projects/codel/wiki/Cake> for more information

simplest_tbf.qos:
Simplest possible configuration (TBF): TBF rate limiter with your qdisc attached. TBF may give better performance than HTB on some architectures.

simplest.qos:
Simplest possible configuration: HTB rate limiter with your qdisc attached.

layer_cake.qos:
This uses the cake qdisc as a replacement for both htb as shaper and fq_codel as leaf qdisc. This exercises cake's diffserv profile(s) as different "layers" of priority. This script requires that cake is selected as qdisc, and forces its usage. See: <http://www.bufferbloat.net/projects/codel/wiki/Cake> for more information

simple.qos:
BW-limited three-tier prioritisation scheme with your qdisc on each queue. (default)

Show and Use Advanced Configuration. Advanced options will only be used as long as this box is checked.

Add

Save & Apply Save Reset

Kuva 3. SQM Queue Discipline -välilehti

3.5 CoDel

CoDel on AQM-algoritmi, joka suunniteltiin helposti käytettäväksi ja adaptoitavaksi ratkaisuksi reitittimien puskuriin verkkopaketeista muodostuvan jonon hallinnoimiseen.⁵ Tavoitteena CoDel-algoritmille oli saada yleiseen jakoon kaiken tasoisiin verkkolaitteisiin sopiva jononhallintajärjestelmä, joka julkaistiin vuonna 2012 Van Jacobsonin ja Kathleen Nicholisin toimesta ja sisällytettiin osaksi Linuxin kerneliä.

CoDel-algoritmille annetaan kolme parametria, limit, target ja interval, jonka avulla se hallinnoi reitittimen puskuriin muodostuvaa verkkopakettien jonoa. Limit-parametri määrittelee, kuinka pitkä reitittimen puskuriin muodostuva jono

saa olla pisimillään ja jonoon muodostuvien verkkopakettien määrän ylittäessä tämän rajan, kaikki uudet saapuvat paketit tiputetaan. Target-parametri määrittelee kuinka pitkään verkkopaketti saa olla jonossa ajallisesti. interval on jonon tarkastelujakson pituus millisekunteina. Algoritmi merkitsee kaikki jonoon saapuvat verkkopaketit aikaleimalla ja se tarkastelee annetun tarkastelujaksonpituisella ajalla, ylittääkö jono pituusrajoituksen sekä pysyvätkö jonossa olevat paketit tarkastelujakson ajan target-parametriin määritellyn maksimi-jonotusajan alla. Mikäli verkkopaketit jonottavat alle annetun viivearvon tarkastelua jakson ajan, paketteja ei merkitä tiputettavaksi. Sen sijaan, jos jonossa olevan verkkopaketin jonotusaika pysyy yli target-parametrin määrittelemän raja-arvon koko tarkastelujakson ajan, CoDel siirtyy tiputustilaan ja alkaa merkitsemään paketteja tiputettavaksi algoritmin siirtyessä Dequeue-rutiiniin. Dequeue-rutiini on funktio, jolla algoritista vapautetaan paketteja siirtymään kohti määränpäättään FIFO-menetelmällä. Tiputustilassa CoDel merkitsee jonosta poistuvia paketteja tiputettavaksi, kunnes jonossa olevien pakettien jonotusaika palaa taas takaisin alle target-parametrin raja-arvon tarkasteluajan pituiseksi ajaksi.^{13 14}

3.6 FQ-CoDel

FQ-CoDel on kehitetty verkkopakettien vuorontamisesta ja aktiivisesta jononhallinnomisesta muodostuva yhdistelmä, joka jatkokehitettiin CoDel-algoritmin pohjalta. FQ-CoDel-algoritmin kehitti Eric Dumazet yhteistyössä bufferbloat.com -yhteisön kanssa muutama vuosi alkuperäisen CoDel-algoritmin julkaisun jälkeen.¹⁵

¹³ Nichols, K. Jacobson, V. McGregor, A. & Iyengar, J. 2018. Controlled Delay Active Queue Management

¹⁴ Ubuntu Manpage Repository. 2022. CoDel.

¹⁵ Bufferbloat.net. 2022. CoDel Overview

Sen sijaan, että reitittimelle saapuvat datavirrat käsitellään yhtenä jonona, FQ-CoDel jakaa reitittimelle saapuvien datavirtojen verkkopaketit erillisiin jonoihin pakettien otsikkotiedoista laskettavan hajautusarvon perusteella. Laskettava hajautusarvo muodostuu verkkopaketin otsikkotietojen lähde- ja kohdeosoitteesta, lähde- ja kohde porttinumerosta sekä kuljetusprotokollan tunnistenumera. Reitittimen kautta kulkevien datavirtojen jakaminen erillisiin käsiteltäviin jonoihin mahdollistaa tiedonsiirtokapasiteetin jakamisen tasaisesti niiden välillä. Tiedonsiirtokapasiteetin jakaminen tasaisesti jonojen välille estää isompia tiedonsiirtoja dominoimasta verkkoliikenneyhteyttä. Jokaista jonoa hallinnoi CoDel-algoritmi, joka noudattaa FIFO-käytäntöä ja käyttää edellisessä luvussa läpikäytyjä parametreja verkkopakettien jonotus ajanhallintaan.¹⁶

Algoritmin vuorontaja jakaa datavirrat sille annettavan flows-parametrin mukaiseen määrään jonoja. Jonojen pituutta mitataan jonoissa olevien pakettien määrän mukaan ja tätä rajoittaa limit-parametri. Jonon saavuttaessa limit-parametrin määrittelemän maksimipituuden, jonosta tiputetaan sen kärjessä olevia paketteja ja saadaan tilaa uusille jonoon saapuville paketeille. Jokainen jono saa quantum-parametrin verran krediittejä, jotka määrittelevät kuinka monen tavun verran dataa jonosta voidaan vapauttaa verkkopaketteina etenemään kohti määränpäättään.¹⁶

Jonot jaetaan kahteen tasoon, uudet ja vanhat, joita vuorontaja käy läpi kierroskierrokselta. Vuorontaja käy ensin läpi uusien jonojen listan ja valitsee siinä ensimmäisenä olevan jonon. Mikäli jonolla ei ole krediittejä tai negatiivinen määrä, sille annetaan quantum-arvon verran krediittejä lisää ja se siirretään vanhojen jonojen

¹⁶ Høiland-Joergensen, T. McKenney, P. Täht, D. Gettys, J. & Dumazet, E. 2018. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm.

listan viimeiseksi. Vanhat jonot, joihin saapuu verkkopaketteja ja joille lisätään krediittejä vuorontajan toimesta, pysyvät vanhojen jonojen listalla.¹⁶

Vuorontaja siirtyy tyhjentämään jonoa, jolla on krediittejä ja verkkopaketteja odottamassa. Tyhjennykseen valittu jono käsitellään CoDel-algoritmin avulla ja siitä vapautetaan quantum-parametrin verran verkkopaketteja tai tiputetaan paketteja, mikäli jonossa olevien verkkopakettien jonotusaika ei pysy alle CoDel-algoritmin target-parametrin määrittelemää raja-arvoa. Vapautettujen verkkopakettien koon verran krediittejä vähennetään jonon krediittisaldosta, kunnes saldo saavuttaa arvon 0 tai alle. Toimenpiteen jälkeen jono siirretään takaisin jonottamaan vanhojen jonojen listalle viimeiselle sijalle.¹⁶

Uusien jonojen listan tyhjennyttyä vuorontaja siirtyy käymään läpi vanhojen jonojen listalla olevia jonoja. Vanhojen jonojen listalla olevien jonojen läpikäynti toimii samalla periaatteella kuin uusien jonojen käsittely jonojen pysyessä aktiivisena. Kun CoDel-algoritmi ei onnistu vapauttamaan yhtään pakettia jonosta, se merkitään tyhjäksi. Mikäli tyhjä jono on uusien jonojen listalla, se siirretään vanhojen jonojen listan viimeiselle sijalle. Tyhjä jono vanhojen jonojen listalla vuorostaan vapautetaan uuteen käyttöön ottamaan vastaan uuden datavirran verkkopaketteja.¹⁶

3.7 CAKE

CAKE on verkkojonon hallintajärjestelmä, joka on vuorostaan jatkokehitetty FQ-CoDelin pohjalta. CAKE julkaistiin vuonna 2018 Dave Tähtin sekä CAKE-järjestelmän luojan Jonathan Mortonin toimesta. Tämä bufferbloattia vastaan kehitetty ratkaisu on suunniteltu kuluttajamarkkinoille suunnatuille reitittimille OpenWrt-

ohjelmistojärjestelmän SQM-paketin kautta luvaten erinäisiä etuja FQ-CoDeliin verrattuna.¹⁷

CAKE eroaa FQ-CoDelista tarjoamalla verkon kaistanleveyden hallinnan, käyttämällä uutta Cobalt-algoritmia jonojen hallintaan, antamalla mahdollisuuden luokitella verkkopaketeissa liikkuvaa dataa ja lisäämällä verkkoliikenteeseen mahdollisuuden suodattaa pois TCP-yhteyksissä esiintyviä turhia ACK-paketteja.¹⁷

Kaistanleveyden hallinta tarjoaa mahdollisuuden rajata reitittimeltä ulkoverkkoon lähtevän datan määrää. Rajaamalla siirrettävän datan määrää, estetään verkkoliikenteeseen muodostuvan pullonkaulan syntymistä käyttäjän verkon päässä. Kuten SQM-järjestelmästä kertovassa kappaleessa käytiin läpi, niin siirtonopeudet yleensä asetetaan 85–90 % maksimikaistanleveydestä, vaikka CAKE lupaa, jopa 99 % varauksen olevan mahdollista.¹⁸

Cobalt-algoritmi koostuu CoDel- ja BLUE-algoritmin yhdistelmästä. Tämän algoritmien yhdistelmä hallinnoi verkkopakettijonoja CoDel-algoritmilla lisäten sen tapaan reagoida jonoihin muodostuvaan viiveeseen dynaamisemmin. CoDel-algoritmi reagoi jonoihin muodostuvaan viiveeseen käynnistämällä pakettien tiputus-tilan ja pysyy tipustus-tilassa, kunnes jonoissa ilmaantuva viive palautuu sen raja-arvojen sisälle. CoDel hallinnoi verkkopakettien tiputtamisen tiheyttä alla esitetyn kaavan 1 mukaisesti. Kaavassa *curTime* indikoi tämänhetkistä aikaa, *nextDropTime* kertoo ajanhetken, milloin seuraava verkkopaketti tulisi tiputtaa, *interval* on jonossa ilmaantuvan viiveen tarkasteluväli ja *count* määrittelee kuinka monta pakettia algoritmin tulisi tiputtaa kyseisenä ajankohtana.¹⁹

¹⁷ Høiland-Jørgensen, T. Täht, & D. Morton, J. 2018. Piece of CAKE: A Comprehensive Queue Management Solution for Home Gateways.

¹⁸ Corbet, J. 2018. Let them run CAKE.

¹⁹ Palmei, J. Gupta, S. Imputato, P. Morton, J. Tahiliani, M. P. Avallone, & S. Täht, D. 2019. Design and Evaluation of COBALT Queue Discipline.

$$nextDropTime = curTime + \frac{interval}{\sqrt{count}} \quad (1)$$

CoDel-algoritmin verkkopakettien tiputustiheys kasvaa aina lineaarisesti algoritmin ollessa tiputustilassa ja kasvu pysähtyy kuin seinään sen päättyessä, koska algoritmin poistuessa tiputustilasta count-arvo palautetaan nolnaan. Tämä johtaa siihen, että jonojen kehittäessä paljon viivettä CoDel-algoritmi joutuu siirtymään jatkuvasti tiputustilaan ja takaisin kasvattaen aina hitaasti verkkopakettien tiputustiheyttä tehden siitä tehottoman runsaasti viivettä kasvattavien jonojen hallintaan. Sen sijaan Cobalt-algoritmissa tämä ratkaistaan pitämällä jonon tiputustilan count-arvo muistissa ja aloittamalla seuraavan tiputustilan tiputustiheys tästä arvosta nollan sijaan. Tämä tekee Cobalt-algoritmin tavasta hallinnoida jonoissa ilmaantuvaa viivettä tehokkaampaa kuin mitä se on CoDel-algoritmillä.¹⁹

Toinen tapa, jolla Cobalt-algoritmi eroaa CoDelista, on sen tyyli hallinnoida jonoja kahdella toisistaan riippumattomalla algoritmilla rinnakkain. CoDel-algoritmin lisäksi Cobalt käyttää BLUE-algoritmia merkitsemään jonoihin saapuvia verkkopaketteja tiputettavaksi, kun jonoon saapuvien pakettien määrä kasvaa ja jonosta alkaa tippumaan paketteja. BLUE-algoritmin logiikka perustuu yhteen todennäköisyysarvoon nimeltään pdrop, jonka mukaan algoritmi merkitsee jonoon saapuvia verkkopaketteja tiputettavaksi. Mikäli jono tiputtaa jatkuvasti paketteja pdrop-arvoa kasvatetaan, mikä lisää todennäköisyyttä, että algoritmi merkitsee uusia paketteja tiputettavaksi niiden saapuessa jonoon. Jonon pysyessä tyhjänä tai datavirran lähteen pysyessä epäaktiivisena, pdrop-arvoa vuorostaan pienennetään. BLUE-algoritmin pdrop-arvo määritellään kolmen liitännäisen parametrin mukaan: freeze_time, increment ja decrement. Ensimmäinen parametri freeze_time kertoo minimiaikavälin kahden onnistuneen pdrop-arvon päivityksen välillä, joka antaa pakettien merkitsemistodennäköisyydelle aikaa vaikuttaa jonon hallintaan ennen seuraavaa pdrop-arvon päivitystä. Parametrin oletusarvona on 100 ms. increment-parametri on oletusarvoltaan 0.00025, joka lisätään pdrop-arvoon, kun

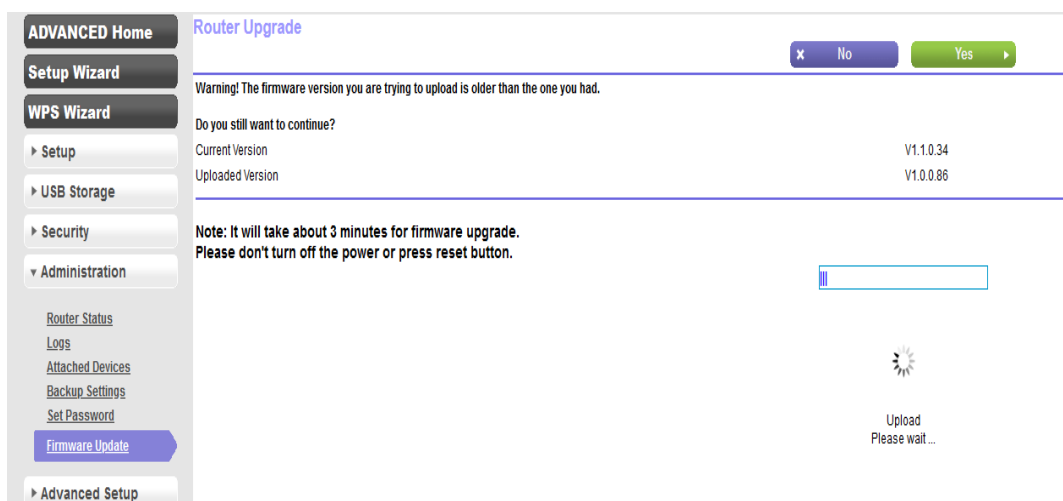
jonoissa tapahtuu ylivuotoa kasvattaen jonoon saapuvien pakettien todennäköisyyttä tulla merkityksi tiputettavaksi ja decrement-arvo vuorostaan vähennetään jonon tyhjetessä.¹⁹

CAKE käyttää DiffServ -protokollaa merkitsemään reitittimen käsittelemien verkkopakettien QoS-luokituksen. DiffServ -luokitus tehdään verkkopakettien otsikkotiedoissa. Tämä luokitus toimii samalla tavalla kuin Netgearin QoS-ominaisuus, joka mahdollistaa tiettyjen pakettien luokittelamisen tärkeämmiksi ja toisten matalamman luokan bulkkitarvikkeiksi. CAKE-järjestelmän kaistanhallinta varaa tärkeämmille paketeille vähän enemmän kaistaa kuin muille verkkopaketeille.¹⁷

Viimeinen CAKE-järjestelmän lisäys on ACK-filteröinti, joka tiputtaa tarpeettomat ACK-paketit TCP-verkkoliikenteestä. CAKE-järjestelmä tarkastelee jonoon saapuvia verkkopaketteja tiputtaen ACK-paketit, jotka eivät sisällä hyödyllistä dataa ja ovat tehty tarpeettomiksi uudempien jonoon asetettujen ACK-pakettien takia, noudattaen TCP-liikenteen sekvenssinumerointia. CAKE saattaa skannata esimerkiksi kolmantena vuorossa olevan ACK-paketin, joka sisältää 10 000 tavua dataa, jonka perässä on tulossa samainen paketti, jossa on vuorostaan 20 000 tavua dataa. Tässä tapauksessa kyseisen 10 000 tavun paketin voi huoletta tiputtaa ja tämä mahdollistaa paremman tiedonsiirtotahdin TCP-yhteyden välille säästämällä tiedonsiirtokapasiteettia karsimalla turhia verkkopaketteja.¹⁸

4 OPENWRT-LAITEOHJELMISTON ASENNUS

OpenWrt-laiteohjelmiston asennus aloitettiin lataamalla laiteohjelmisto OpenWrt-sivustoilta Netgear R6220 -reitittimelle luodulta sivulta, joka sisältää tietoa kyseisestä reitittimestä, muutamia huomioitavia asioita liittyen kyseisen laiteohjelmiston sekä R6220 -reitittimen toimintaan ja asennusohjeet. Laiteohjelmiston asennus tapahtuu yksinkertaisesti Netgear -reitittimen laiteohjelmiston päivityssivun kautta syöttämällä OpenWrt-sivustolta ladattu laiteohjelmistotiedosto päivityssivulle, jonka jälkeen päivitys käynnistetään ja päivitysprosessi hoitaa uuden laiteohjelmiston asennuksen automaattisesti alla olevan kuvan 5 mukaisesti.



Kuva 4. OpenWrt-laiteohjelmiston asennus Netgear-laiteohjelmistopäivityksen avulla

Laiteohjelmiston asennuksen jälkeen reititin käynnistyy uudelleen ja reititin näyttää OpenWrt-laiteohjelmiston sisäänkirjautumissivun, johon syötetään oletus sisäänkirjautumistunnukset. Reitittimeen kytketyille koneille määriteltiin staattiset IP-osoitteet laiteohjelmiston ”DHCP and DNS”- asetussivulla. Kuvassa 6 näytetään reitittimiseen liitettyjen Desktop, P9-1 ja P1-2-laitteille asetetut staattiset IP-osoitteet loputtomalla leasing ajalla, jotta laitteiden IP-osoitteet pysyvät samana koko testauksen ajan.

DHCP and DNS

Dnsmasq is a lightweight DHCP server and DNS forwarder.

General Settings | **Resolv and Hosts Files** | PXE/TFTP Settings | Advanced Settings | **Static Leases** | Hostnames | IP Sets

Static leases are used to assign fixed IP addresses and symbolic hostnames to DHCP clients. They are also required for non-dynamic interface configurations where only hosts with a corresponding lease are served. Use the *Add* Button to add a new lease entry. The *MAC address* identifies the host, the *IPv4 address* specifies the fixed address to use, and the *Hostname* is assigned as a symbolic name to the requesting host. The optional *Lease time* can be used to set non-standard host-specific lease time, e.g. 12h, 3d or infinite.

Hostname	MAC address	IPv4 address	Lease time	DUID	IPv6 suffix (hex)	
Desktop	D4:5D:64:53:D7:73	10.0.2.2	infinite	000100012a250ce8d45d6453d773	none	<input type="button" value="≡"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
P9-1	48:0F:CF:46:C8:A8	10.0.2.3	infinite	none	none	<input type="button" value="≡"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
P1-2	48:0F:CF:3D:B2:84	10.0.2.4	infinite	none	none	<input type="button" value="≡"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Active DHCP Leases

Hostname	IPv4 address	MAC address	Lease time remaining
P1-2 (PK3_EC3_P1-2.lan)	10.0.2.4	48:0F:CF:3D:B2:84	unlimited
P9-1 (P9-1.lan)	10.0.2.3	48:0F:CF:46:C8:A8	unlimited
Desktop (Desktop.lan)	10.0.2.2	D4:5D:64:53:D7:73	unlimited

Kuva 5. Reitittimeen kytkettyjen koneiden staattiset IP-osoitteet ja leasing ajat OpenWrt-laiteohjelmiston DHCP/DNS-asetussivulla

Laiteohjelmiston asetusten määrittelemisen viimeisessä vaiheessa asennetaan SQM-ohjelmistopaketti. SQM-ohjelmistopaketti löytyy OpenWrt-laiteohjelmiston "Software" -asetussivulla hakemalla nimikkeellä "luci-app-sqm". Ohjelmistopaketin asennus vaatii vain Install -painikkeen painamista ja OpenWrt tarjoaa asennukseen liittyvät tiedot erillisellä ikkunalla kuvan 7 mukaisesti. Ikkunassa esitetään ohjelmistopaketin kokoon ja liitännäisten osien pakettien nimiä. SQM-ohjelmistopaketin asennuksen jälkeen reititin oli valmis testauksia varten OpenWrt-laiteohjelmistolla.

OpenWrt Status

Software

Free space: 99% (17.9 MB)

Filter: sqm

Package name	Version	Status
sqm-scripts	1.5.2-	NOT INSTALLED
sqm-scripts-extra	2016-4	NOT INSTALLED
collectd-mod-sqm	5.12.0	NOT INSTALLED
luci-app-sqm	git-21	NOT INSTALLED
luci-i18n-sqm-ar	git-22	NOT INSTALLED
luci-i18n-sqm-bg	git-22	NOT INSTALLED
luci-i18n-sqm-bn	git-22	NOT INSTALLED
luci-i18n-sqm-ca	git-22	NOT INSTALLED
luci-i18n-sqm-cs	git-22	NOT INSTALLED
luci-i18n-sqm-da	git-22	NOT INSTALLED
luci-i18n-sqm-de	git-22	NOT INSTALLED
luci-i18n-sqm-el	git-22	NOT INSTALLED

Details for package *luci-app-sqm*

Version: git-21.164.25180-005c09a
 Size: ~3.4 KB installed

Dependencies:

- ↳ sqm-scripts (18.1 KB) NOT INSTALLED
- ↳ tc-bpf (143.1 KB) NOT INSTALLED | tc-full (142.8 KB) NOT INSTALLED | tc-tiny (135.5 KB) NOT INSTALLED
- ↳ kmod-sched-core (64.2 KB) NOT INSTALLED
- ↳ kernel INSTALLED
- ↳ libmnl0 INSTALLED
- ↳ libbpf20220308 (135.4 KB) NOT INSTALLED
- ↳ libelf1 (37.1 KB) NOT INSTALLED
- ↳ zlib (37.4 KB) NOT INSTALLED
- ↳ libxtables12 (18.3 KB) NOT INSTALLED
- ↳ tc-mod-iptables (3.4 KB) NOT INSTALLED
- ↳ kmod-sched-cake (18.2 KB) NOT INSTALLED
- ↳ kmod-ibf (3.3 KB) NOT INSTALLED
- ↳ iptables-nft (247 B) NOT INSTALLED | iptables-zz-legacy (250 B) NOT INSTALLED
- ↳ kmod-ipt-core (11.2 KB) NOT INSTALLED
- ↳ kmod-nf-reject INSTALLED
- ↳ kmod-nf-ipt (20.1 KB) NOT INSTALLED
- ↳ kmod-nf-log INSTALLED
- ↳ xtables-nft (58.7 KB) NOT INSTALLED
- ↳ libnftnl11 INSTALLED
- ↳ libiptext0 (32.4 KB) NOT INSTALLED
- ↳ libiptext6-0 (9.3 KB) NOT INSTALLED
- ↳ libiptext-nft0 (2.7 KB) NOT INSTALLED
- ↳ kmod-nft-compat (4.6 KB) NOT INSTALLED
- ↳ kmod-nft-core INSTALLED
- ↳ kmod-nfnetlink INSTALLED
- ↳ kmod-nf-reject6 INSTALLED
- ↳ kmod-nf-contrack6 INSTALLED
- ↳ kmod-nf-contrack INSTALLED
- ↳ kmod-nf-nat INSTALLED
- ↳ kmod-nf-log6 INSTALLED
- ↳ kmod-lib-crc32c INSTALLED
- ↳ kmod-crypto-crc32c INSTALLED
- ↳ kmod-crypto-hash INSTALLED
- ↳ xtables-legacy (21.5 KB) NOT INSTALLED
- ↳ libip4tc2 (9.7 KB) NOT INSTALLED
- ↳ libip6tc2 (9.8 KB) NOT INSTALLED
- ↳ iptables-mod-ipopt (12.7 KB) NOT INSTALLED
- ↳ kmod-ipt-ipopt (6.1 KB) NOT INSTALLED

Description

Luci interface for the SQM scripts queue management package

Require approx. 959.4 KB size for 27 package(s) to install.

Overwrite files from other package(s)

Cancel Install

Kuva 6. SQM-ohjelmistopakettien asennusikkuna tietoineen

5 LIALLISEN PUSKUROINNIN TESTAAMINEN

5.1 Laitteet

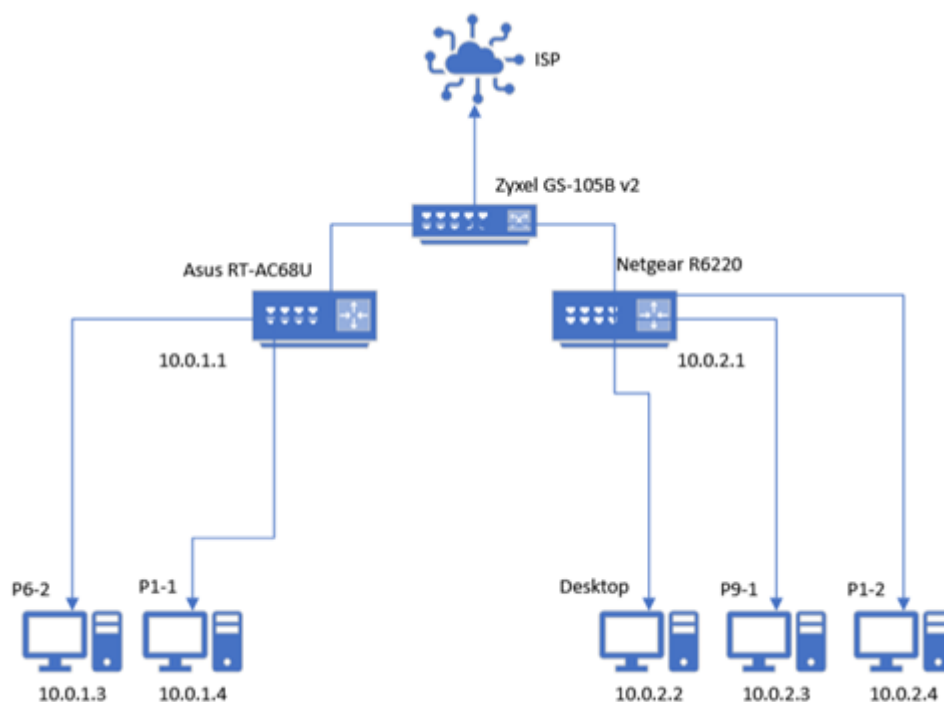
Testauksissa käytettiin viittä tietokonetta, kahta reititintä ja yhtä kytkintä. Tietokoneissa käytettiin Windows 10 -käyttöjärjestelmää ja koneet kytkettiin verkkoon koneiden emolevyssä olevaan verkkokorttiin, jotka olivat kykeneviä 1 Gbps tiedonsiirtonopeuteen. Reitiminä toimivat Asus RT-AC68U ja Netgear R6220, joista jälkimmäinen oli testattava laite. Kytkimenä käytettiin Zyxel GS-105B v2:sta, joka on yksinkertainen viisiporttinen Gigabitin verkkokytkin.

5.2 Ohjelmisto

Testausten viive mitattiin Windows-käyttöjärjestelmän komentokehötteen ping-komennolla ja tiedonsiirtoliikenne luotiin iPerf3-ohjelmalla. Reitittimet oli päivitetty viimeisimpiin laiteohjelmistoversioihin: Asus RT-AC68U-version 3.0.0.4.386.49703 ja Netgear R6220 version 1.1.0.34. Netgear -reitittimeen asennettu OpenWrt-laiteohjelmiston versio oli myös viimeisin julkaisu 22.03.2. OpenWrt-laiteohjelmistossa käytetyn SQM-paketin version oli git-21.164.25180-005c09a.

5.3 Kytkennät

Reitittimet jaettiin eri verkkoihin kytkimen avulla muodostaen molemmille reititimille oman julkisen IP-osoitteen. Kytkin liitettiin ulkoverkkoon yksittäisellä Ethernet-kaapelilla. Molempiin reitittimiin asetettiin DHCP:n avulla oma lähiverkko ja staattiset IP-osoitteet kytketyille tietokoneille selkeyttämään verkkotopologiaa. Kytkettyjen laitteiden topologia näkyy kuvassa 8. Laitteet kytkettiin toisiinsa CAT6 Ethernet-kaapeleilla.



Kuva 7. Laitekytkentöjen topologia

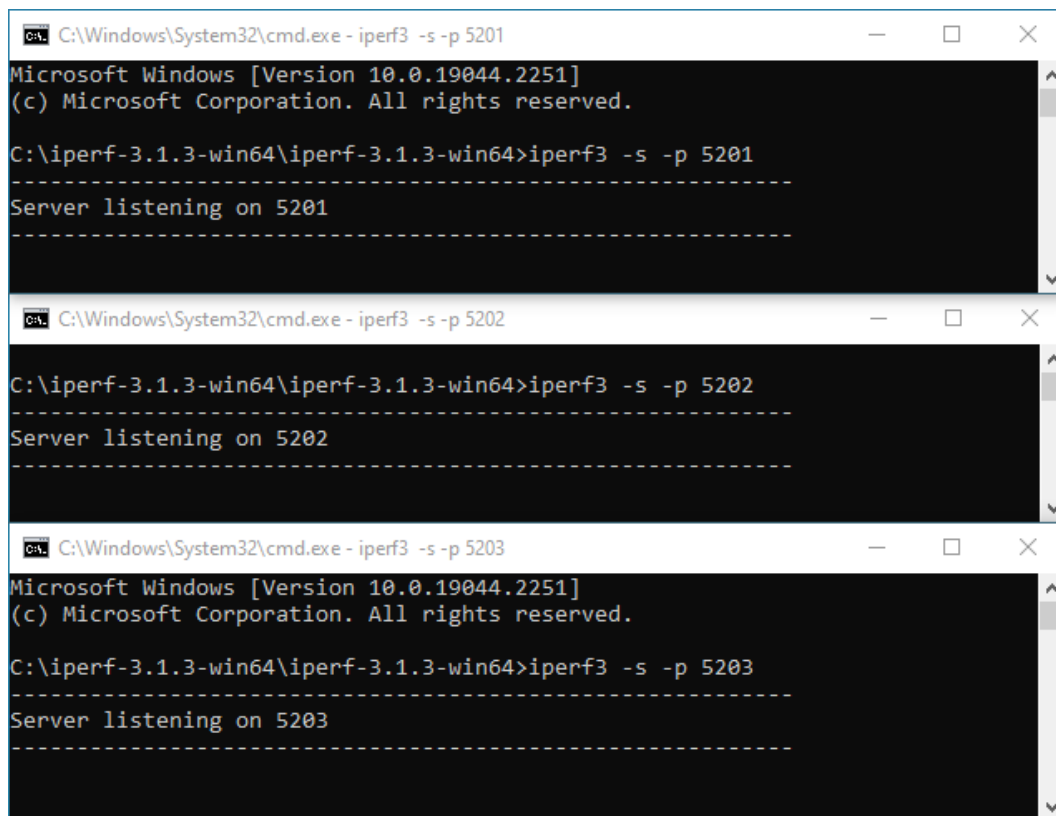
5.4 iPerf3

iPerf3-ohjelmalla oli tarkoitus tuottaa testauksissa verkkoyhteyttä kuormittavaa verkkoliikennettä. Mittauksissa Asus-reitittimeen kytketyt koneet toimivat iPerf3 -palvelimina ja Netgear-reitittimeen kytketyt P9-1- ja P1-2-koneet toimivat asiakasohjelmina. Kaikilla neljällä koneella käynnistettiin kolme erillistä iPerf3-ohjelmaa. Jakamalla iPerf3-datavirrat erillisiin prosesseihin mahdollisesti testauskoneilla 45 rinnakkaisen datavirran jakamisen 15 rinnakkaiseen datavirtaan per suoritin- ydin johtuen ohjelman yksisäikeisyydestä.²⁰ Verkkoyhteyteen generoidun kuorman maksimointi kokeiltiin suorittamalla testejä, joissa mitattiin verkkoyhteydessä muodostuva viive suhteessa datavirtojen määrään ja datavirtojen jakaminen kolmeen osaan mahdollisesti täyden väylätaajuuden saavuttamisen koneilla.

²⁰ Dell.com. 2022. PowerScale: iperf3-käyttö OneFS:ssä.

Kolmen erillisen datavirran ylittäminen ei saavuttanut suurempaa viivettä testatuilla verkkoyhteys nopeuksilla.

Palvelinohjelmat käynnistettiin kuvan 9 tapaisesti molemmilla Asus-reitittimeen kytketyillä koneilla. Komennossa "-s" -parametri viittaa serveriin ja "-p 5201" -porttiin, joka prosessille avattiin tiedonsiirtoa varten.



```
C:\Windows\System32\cmd.exe - iperf3 -s -p 5201
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\iperf-3.1.3-win64\iperf-3.1.3-win64>iperf3 -s -p 5201
-----
Server listening on 5201
-----

C:\Windows\System32\cmd.exe - iperf3 -s -p 5202
C:\iperf-3.1.3-win64\iperf-3.1.3-win64>iperf3 -s -p 5202
-----
Server listening on 5202
-----

C:\Windows\System32\cmd.exe - iperf3 -s -p 5203
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\iperf-3.1.3-win64\iperf-3.1.3-win64>iperf3 -s -p 5203
-----
Server listening on 5203
-----
```

Kuva 8. iPerf3-palvelin käynnistyskomento

Vastaavasti P9-1- ja P1-2-koneilla käynnistettiin kolme iPerf3-asiakasohjelmaa, jotka aloittavat saman tien verkkopakettien lähettämisen palvelinohjelmiin komennon suoritettua. Asiakasohjelman käynnistyskomento näkyy kuvassa 10. Kuvassa näkyy komento asiakasohjelman käynnistämiseen ja käynnissä oleva tiedonsiirto toisessa verkossa olevaan palvelinohjelmaan.

```

C:\Windows\System32\cmd.exe - iperf3 -c 88.115.148.91 -p 5202 -t 9999 -P 15
[ 12] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 14] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 16] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 18] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 20] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 22] 2611.00-2612.01 sec 128 KBytes 1.05 Mbits/sec
[ 24] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 26] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 28] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[ 30] 2611.00-2612.01 sec 128 KBytes 1.05 Mbits/sec
[ 32] 2611.00-2612.01 sec 256 KBytes 2.09 Mbits/sec
[SUM] 2611.00-2612.01 sec 3.38 MBytes 28.2 Mbits/sec

```

Kuva 9. iPerf3-asiakasohjelma

Komennossa parametri `”-c 88.115.148.91”` tarkoittaa ohjelman käynnistämistä asiakasroolissa ottaen yhteyttä kyseisen IP-osoitteen takana olevaan palvelinohjelmaan, `”-t 9999”` -parametri vuorostaan määrittelee, kuinka monta sekuntia komentoa suoritetaan ja `”-P 15”` kertoo ohjelmalle, että sen tulee käynnistää 15 rinnakkaista datavirtaa. P1-2-koneella komentoon lisättiin `”-R”` -parametri, joka asettaa asiakasohjelman vastaanottamaan verkkopaketteja palvelinohjelmalta lähettämisen sijaan.

5.5 Testausmenetelmä

Testaukset jaettiin kahteen osaan. Ensimmäisessä osassa mittaukset suoritettiin Netgear -reitittimen oletuslaiteohjelmistolla ja toisessa osassa samat mittaukset suoritettiin uudestaan, kun Netgear -reitittimeen asennettiin OpenWrt-laiteohjelmisto SQM-ohjelmapaketin kera. Testauksissa mitattiin reitittimen tiedonsiirtonopeus ilman verkkoliikennekuormitusta ja verkossa esiintyvää latenssia mitattiin eri reititinasetuksin kuormittuneessa ja kuormittumattomassa verkkoyhteydessä. Ensin mitattiin miten raskas tiedonsiirto verkon kautta vaikuttaa verkon latenssiin ja tämän jälkeen kokeiltiin QoS -järjestelmällä verkkoyhteydessä muodostuvan viiveen vähentämistä ja tämän vaikutusta verkon nopeuteen. OpenWrt-järjestelmällä mittaukset toistettiin verrataksemme laiteohjelmistoja ja niiden tarjoamia verkkoliikenteen hallintamenetelmiä. Testeissä käytettiin kahta eri Elisan verkkoliittymää. Verkkoliittymät olivat Elisan Netti Kotiin XXL-liittymä ja M-liittymä. XXL-liittymä kykeni keskimäärin noin 900+ Mbit/s lataus- ja 340+ Mbit/s lähetysnopeuksiin testien aikana. M-liittymän nopeudet olivat keskimäärin 140+/40+

Mbit/s. Tiedonsiirronnopeutta kuvaavat yksiköt ovat megabittejä sekunnissa ja testeissä käytettyjen verkkoliittymien luvatut arvot näkyvät kuvassa 11.

Elisa Netti kotiin XXL

- ✓ Lisäpalveluina saatavilla yhteistyössä F-Securen kanssa toteutettu Elisa Turvapaketti ja kotimainen pilvitalennuspalvelu Elisa Pilvilinna
- ✓ Toteutuviin yhteysnopeuksiin vaikuttavat muun muassa valittu palvelu, käytettävä verkkotekniikka, kiinteistön sisäverkko, häiriötaso, päätelaitteen ominaisuudet ja verkon kuormitus. Yhteysnopeuden vaihteluvälit (Mbit/s): Lataus 1000M 500-1000, normaalinopeus 900M ja Lähetys 400M 200-400 normaalinopeus 360M.
- ✓ 5 kpl julkisia IP-osoitteita (kiinteä IP-osoite ei saatavilla)

Elisa Netti kotiin M

- ✓ Lisäpalveluina saatavilla yhteistyössä F-Securen kanssa toteutettu Elisa Turvapaketti ja kotimainen pilvitalennuspalvelu Elisa Pilvilinna
- ✓ Toteutuviin yhteysnopeuksiin vaikuttavat muun muassa valittu palvelu, käytettävä verkkotekniikka, kiinteistön sisäverkko, häiriötaso, päätelaitteen ominaisuudet ja verkon kuormitus. Yhteysnopeuden vaihteluvälit (Mbit/s): Lataus 150M 100-150, normaalinopeus 135M ja Lähetys 50M 35-50 normaalinopeus 45M.
- ✓ 5 kpl julkisia IP-osoitteita (kiinteä IP-osoite ei saatavilla)

Kuva 10. Työssä käytettyjen Elisan laajakaistaliittymien yhteysnopeudet esitettynä Elisan verkkosivuilla.²¹

Netgear-reitittimellä testattiin sen oletus laiteohjelmistolla QoS-ominaisuutta järjestämällä mittauksiin käytettävä Desktop-kone ja verkkoliikennettä generoivat koneet eri tärkeysluokkiin, testaten miten tärkeysluokat vaikuttavat mitattavaan viiveeseen. OpenWrt-laiteohjelmistolla kokeiltiin ensin SQM-järjestelmän Cake -vuorontaja algoritmiyhdistelmällä ja piece_of_cake -skriptillä, käyttäen OpenWrt-sivuston suosittomia 85 % lataus- ja lähetysnopeusrajoituksia. 85 % rajoitusten jälkeen testit toteutettiin 50 % ja 33 % rajoituksilla. Samat testit toistettiin FQ-CoDel vuorontaja-algoritmiyhdistelmällä ja simplest.qos -skriptillä. Testien viimeisessä vaiheessa SQM-järjestelmällä asetettiin FQ-CoDel- ja simplet.qos-yhdistelmä

²¹ Elisa.fi. 2022. Kodin nettiyhteydet

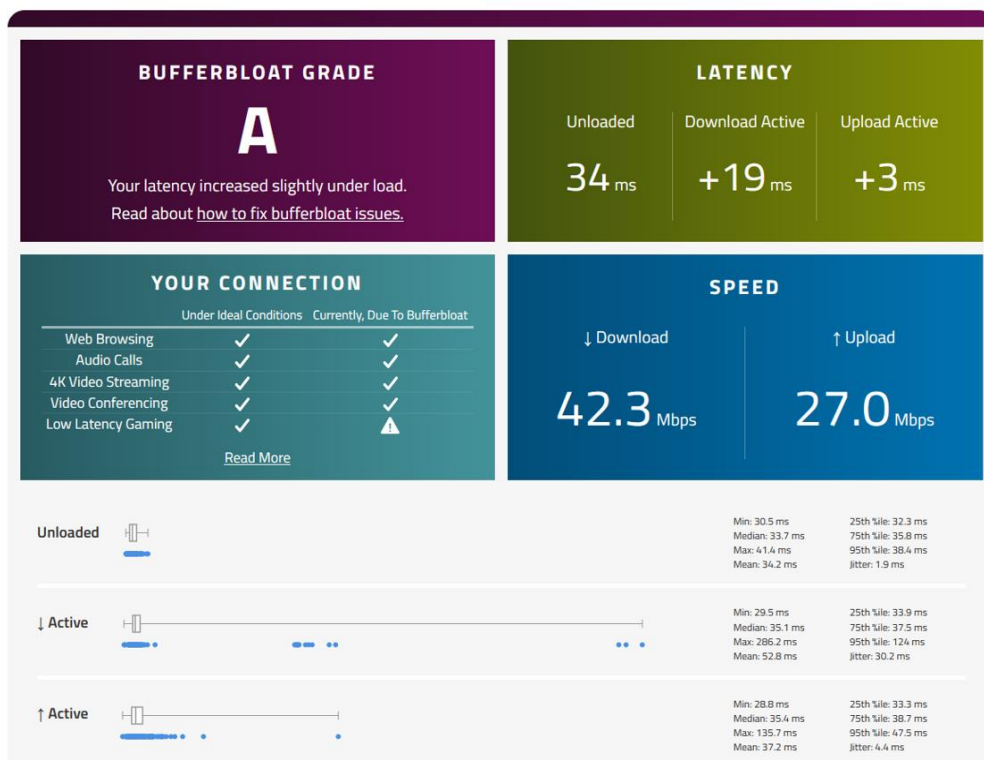
verkkoliikennettä generoivien koneiden verkkoportteille aktiiviseksi, jättäen mittauksia tekevä koneen verkkoportti sekä WAN-portti ilman SQM-järjestelmän hallinnointia.

Mittauksissa viive mitattiin komentokehoteella, jolla lähetettiin ping-komento 100 kertaa Googlen julkiselle DNS-palvelimelle osoitteeseen 8.8.8.8. Ping-komento suoritettiin kolme kertaa ja ping-komentojen latensseista laskettiin keskiarvo, joka muodosti lopullisen mittaustuloksen. Viive mitattiin ensin verkkoyhteydessä, johon ei generoitu ylimääräistä verkkoliikennettä, ja tätä mittaustulosta verrattiin verkkoyhteydestä saatuun tulokseen, johon generoitiin liikennettä iPerf3-ohjelmalla. Tuloksissa esiintyvät tiedonsiirtonopeudet mitattiin kolmella eri verkkonopeutta mittaavalla sivulla kerran, joista laskettiin keskiarvo verkon lataus- ja lähetysnopeuksille eri testien aikana. Testaukseen käytettiin Ookla SpeedTest (speedtest.net), Speed Test Group- (speedtestgroup.com/#speed-test) ja testmy.net (testmy.net) -sivustojen tarjoamia verkon nopeustestejä.

Verkkoliikennemittauksiin muodostettiin iPerf3-ohjelmalla, joka mahdollistaa datavirtojen muodostamisen lähiverkkoihin kytkettyjen koneiden välille. Mittauksissa P9-1-koneelta lähetettiin tietoliikennettä P6-2-koneelle ja P1-1-koneelta vuorostaan lähetettiin liikennettä P1-2-koneelle. Tällä asetelmalla Netgear -reitittimen lähiverkkoyhteyteen muodostui sisään- ja ulospäinsuuntautuvaa verkkoliikennettä tavoitteena saada reititin puskuroimaan sen kautta kulkevia verkkopaketteja aiheuttaen viivettä verkkoyhteyteen. Desktop-kone kytkettiin testattavan reitittimen verkkoon erilliseksi verkon viivettä mittaavaksi laitteeksi.

Yhtenä tapana testata verkossa esiintyvää viivettä oli käyttää WaveForm-sivuston Bufferbloat-testiä, mutta tämä testaustapa hylättiin sen antamien tulosten epäluotettavuuden takia. WaveForm-sivuston tulosten epäluotettavuus todennettiin tekemällä sama testi samoilla laiteasetuksilla moneen eri kellonaikaan päivän mittaan. Tulokset vaihtelivat runsaasti toisistaan päivän mittaan, joten sivuston antamiin tuloksiin oli mahdoton luottaa. Kuvassa 12 näytetään esimerkki WaveForm-

sivuston Bufferbloat-testin tuloksesta. Kyseistä testiä on suositeltu käytettäväksi muun muassa OpenWrt-sivustolla.¹²

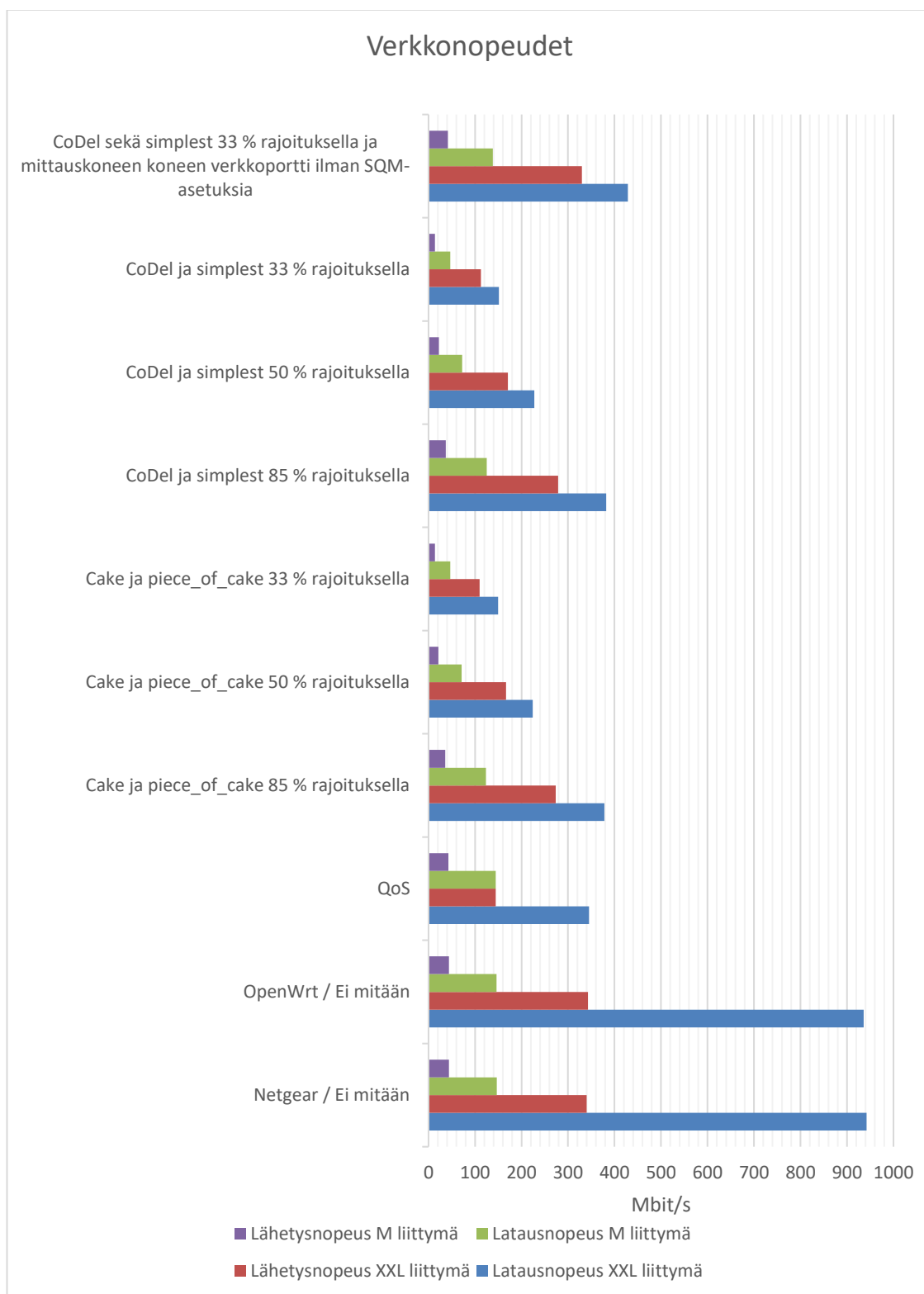


Kuva 11. WaveForm Bufferbloat -testin tulos.²²

²² Waveform.com. 2022. Bufferbloat and Internet Speed Test

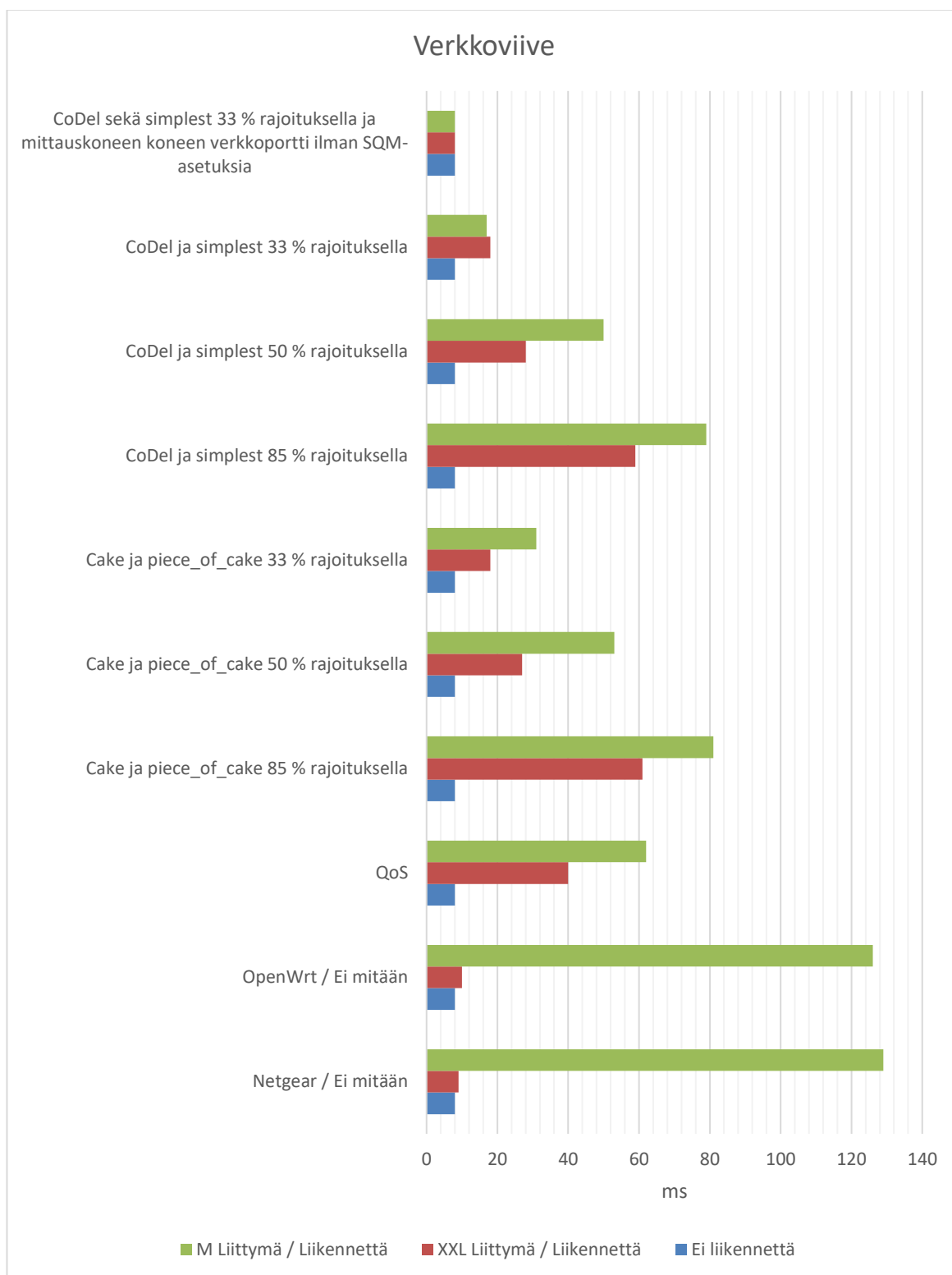
6 TULOKSET

Pylväsdiagrammi kuvassa 13 esittää ensimmäiset reitittimillä mitatut verkon tiedonsiirtonopeudet eri testiasetuksin ja ilman verkkoliikennekuormitusta. Taulukossa tulokset jaotellaan käytetyn liittymäntyyppin, tiedonsiirron kulkusuunnan, käytetyn laiteohjelmiston ja reitittimellä käytettyjen asetusten mukaisesti. Taulukossa "Ei mitään" tarkoittaa, että reitittimessä ei käytetty minkäänlaista liiallisen puskurointiin vaikuttavaa toimintoa mittauksen aikana ja näitä tuloksia on tarkoitus käyttää referenssinä Netgear- ja OpenWrt-laiteohjelmiston verkkopakettien hallintajärjestelmien vaikutukseen verkonsiirtonopeuksiin. Taulukossa CoDel ja Cake merkitsevät testin aikana käytettyä vuorontaja-algoritmiyhdistelmää, simples ja piece_of_cake ilmaisevat mikä Queue setup script oli käytössä testin aikana ja rajoituksilla viitataan verkon lataus- ja lähetysnopeusrajoituksiin, joita SQM-järjestelmän verkkokaistanhallinta käyttää. QoS viittaa Netgear -oletuslaiteohjelmiston verkkopakettien hallintamenetelmään. Taulukon tiedonsiirtonopeuksia kuvaavat luvut ovat megabitteinä sekunnissa.



Kuva 12. Verkon siirtonopeuksia kolmella eri mittauksella, joiden keskiarvosta muodostuu taulukon esittämät tulokset

Toisessa pylväsdiagrammissa (**Kuva 13.**) näytetään verkkoyhteydessä esiintyvä viive mittausten aikana eri testiasetuksin. Taulukko on jaoteltu ensimmäisen taulukon mukaisesti käytetyn laiteohjelmiston, laiteohjelmiston asetusten ja sen mukaan onko verkkoon generoitu verkkoliikennettä mittauksen aikana sekä verkkoliittymä tyyppin mukaan. Taulukossa esitetyt arvot esitetään millisekunteina ja ensimmäisestä taulukosta eroten, tässä diagrammissa pienempi arvo on parempi.



Kuva 13. Verkossa mitatut viiveet eri testiasetuksin kolmella eri mittauksella, joiden keskiarvosta muodostuu taulukossa esiintyvät tulokset

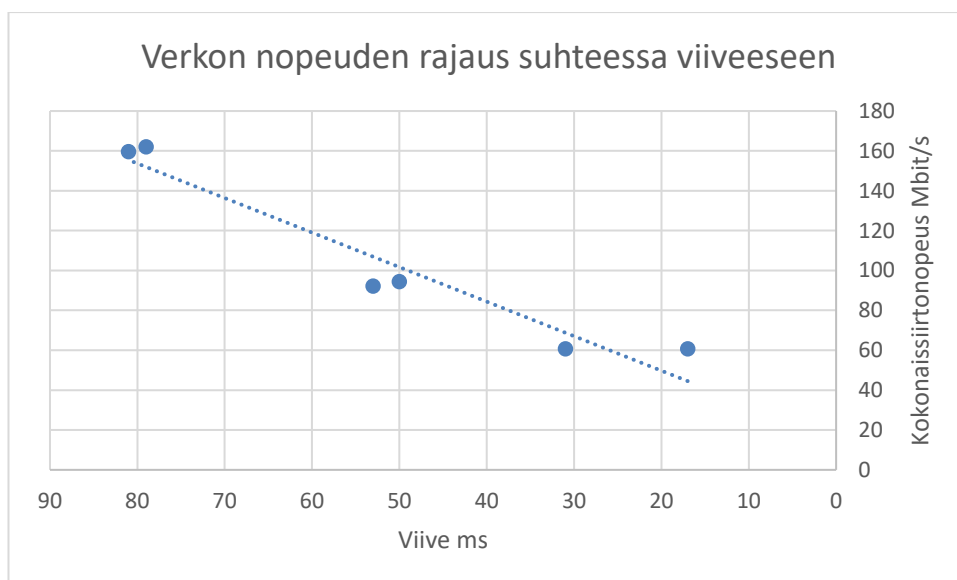
7 TULOSTEN ANALYYSI

Tuloksista voi huomata verkkoyhteyden nopeuden olevan merkittävä tekijä verkossa esiintyvän viiveen muodostumiseen tilanteessa, jossa muut samaan verkkoon kytketyt laitteet luovat paljon verkkoliikennettä. Elisan XXL-liittymän verkkoyhteydessä ei esiintynyt latenssia juuri lainkaan siitä huolimatta, että iPerf3-ohjelmalla generoitiin koko kaistan täydeltä verkkoliikennettä mittausten aikana. Verkkoyhteydessä esiintyvä latenssi osoittautui päinvastaisesti pahenevan, kun reitittimellä käytettiin QoS- tai SQM-ominaisuuksia viiveen vähentämiseksi. Ainoastaan yksi asetus osoittautui antamaan parempia tuloksia kuin verkon käyttö ilman Bufferbloatia ehkäiseviä ratkaisuja ja tälläkin oli hintana verkon nopeuksien puolittuminen. Kyseinen asetus oli SQM-jononhallinnan asettaminen kahden verkkoliikennettä tuottavan koneen LAN-verkkoporttiin rajoittamaan niiden tiedonsiirron määrä kevyimmällä SQM-asetus yhdistelmällä, FQ-CoDel ja `simplest.qos`, jättäen mittauksia tekevän koneen verkkoportin ja WAN-portin reitittimellä ilman SQM-asetuksia. Käyttäen kyseistä asetusta, SQM toimii pääasiassa verkon nopeusrajoittimena verkkoliikennettä tuottaville koneille. Tämä asetus saavutti parhaan viivearvon 8 ms testien aikana, tiputtaen verkon nopeuden puoleen (428/330 Mbit/s) maksimistaan. 8 millisekunnin viiveen saavuttamista ei kuitenkaan voida pitää missään määrin hyvänä tuloksena, kun verkkoyhteys ilman SQM- tai QoS -ominaisuuksia päällä saavutti 9 ms viiveen verkon ollessa kuormitettuna, pitäen verkon latausnopeuden kaksinkertaisena (939 Mbit/s). Yhden millisekunnin ero ei ole havaittavissa käyttäjän näkökulmasta verkon käytettävyyden suhteen.

M-liittymällä latenssia muodostui runsaasti enemmän testausten aikana verrattuna XXL-liittymään ja tähän näytti Netgearin QoS ja OpenWrt-laiteohjelmiston SQM-ratkaisut auttavan. Netgearin QoS pienensi M-liittymän verkkoyhteydessä muodostuvaa viivettä 108 %:sti tuoden viiveen 129 millisekunnista 62:een, päihittäen OpenWrt-ohjelmiston `Cake` ja `piece_of_cake` -asetuksen 53 %:lla. Nämä kaksi tulosta oli parhaiten toisiinsa vertailtavissa, koska ne vaikuttavat yhtä vähän ver-

kon kokonaisnopeuteen. OpenWrt tarjoaa kuitenkin muitakin asetuksia ja mittauksissa ilmeni, että viive pieneni lineaarisesti verkon nopeuden rajauksen kasvun kanssa, kuten kuvan 15 pistekuviosta voidaan havaita. Kuvassa esitetty kokonaissiirtonopeus laskettiin lisäämällä M-liittymän verkon lataus- ja lähetysnopeus yhteen.

Parhaan tuloksen testeissä saavutti jälleen asetus, jossa mittauksia tekevän koneen verkkoportti ja WAN-portti jätettiin ilman SQM-järjestelmän hallinnointia ja rajoittamalla liikennettä tuottavien koneiden verkkoportit. Tämä ratkaisu laski verkon kokonaisnopeutta vain noin 5 %, saavuttaen täydellisen 8 ms viiveen testien aikana.



Kuva 14. Verkon kokonaissiirtonopeus suhteessa mitattuun viiveeseen

Tuloksissa voidaan havainnoida viiveen kasvavan selkeästi, kun reitittimestä tulee pullonkaula tiedonsiirrolle Elisan XXL-liittymässä reitittimen yrittäessä käsitellä SQM-jononhallinnan avulla kaikki sen kautta kulkevat verkkopaketit. Tämä 9 vuotta vanha reititin ei selkeästi kykene käsittelemään 1000/400 Mbit/s nopeuksia kattavien verkkoyhteyksien kaikkia verkkopaketteja, vaikka kyseisen reititinmalli on tuettu uusimmalla OpenWrt-laiteohjelmistoversiolla. Tämä ei kuitenkaan tule täytenä yllätyksenä, koska tästä varoiteltiin OpenWrt-verkkosivuillakin: "SQM is

heavily CPU-based. Slower devices may not be able to keep up with your connection speed”.¹² Jononhallintajärjestelmät eivät ole mikään ihmelääke ikääntyvien verkkolaitteiden performanssin parantamiseen ainakaan, jos haluaa käyttää niitä kaiken reitittimen kautta kulkevan liikenteen käsittelemiseen.

Positiivisena puolena mittauksista saaduista tuloksista voidaan kuitenkin pitää sitä, että OpenWrt-laiteohjelmiston SQM ja Netgearin QOS tarjoavat paremman käytettävyyden Elisan M-tasoiseen verkkoliittymään, jossa esiintyy paljon verkkoliikenteen kuormitusta. Lisäksi parhaan tuloksen tarjonnut asetus, jossa SQM asetetaan liikennettä generoivien koneiden verkkoporttiin aktiiviseksi ja jätetään mittauksia tehneen koneen verkkoportista, sekä WAN-portista pois, ei tunnu olevan ihan kyseisen järjestelmän käyttötarkoituksen mukaista. Käyttämällä SQM-järjestelmää tällä tavalla, se toimii lähinnä verkkoliikenteen nopeuksien rajoittajana. Tästä huolimatta tämä asetus todennettiin tarjoavan M-liittymän tasoisessa verkkoyhteydessä erinomaisen tavan poistaa ainakin yhdeltä koneelta latenssi kokonaan pois sen verkkoyhteydestä, jossa muodostuu paljon verkkoliikennettä, vähentämättä verkonsiirtonopeuksia merkittävästi.

Kyseisistä tuloksissa kannattaa huomioida se, että vaikka tällä laite- ja verkkoyhteys yhdistelmällä nämä asetukset saivat aikaan odotuksen mukaisia tuloksia parantaen verkkoyhteyden laatua, se ei kuitenkaan tarkoita sitä, että niistä voi välttämättä vetää suoria johtopäätöksiä siitä, miten ne tulevat vaikuttamaan jollain toisella laite ja verkkoyhteydyhdistelmällä. Kyseisten ohjelmistoratkaisujen tavoitteena on tuoda verkossa ilmaantuva pullonkaula käyttäjän reitittimelle, jotta verkkopakettien puskurointi sekä hallinnointi tapahtuisi hallitusti, mutta tämä ei välttämättä ole täysin mahdollista kaikilla laitteilla ja verkkoyhteyksillä, tehden tuloksista vain suuntaa antavia.

8 JOHTOPÄÄTÖKSET

Tämän opinnäytetyön tavoitteena oli tutustua Bufferbloat-ilmiöön sekä kyseistä ilmiötä vastaan kehitettyjen ohjelmistoratkaisujen toiminnallisuuteen. Lisäksi mitaamalla liiallista puskurointia vastaan kehitettyjen ratkaisujen vaikutusta verkkonopeuksiin ja verkkoyhteyteen syntyvään latenssiin, voitiin vetää johtopäätöksiä niiden tehokkuudesta kuluttajan näkökulmasta.

Työssä mitattiin verkon tiedonsiirtonopeus puskuroinnin hallitsemiseen kehitettyjen ratkaisujen käytön aikana ja ilman, sekä mitattiin vastaavasti verkkoyhteydessä esiintyvä latenssi, kun verkkoyhteyteen generoitiin runsaasti liikennettä. Tuloksia arvioitiin ja niistä pystyttiin vetämään selkeitä johtopäätöksiä. Työssä esitettyjen tuloksien, johtopäätöksien ja aiheeseen liittyvien termien tietämyksen avulla lukija voi tehdä tietoisin päätöksen, miten lähestyä kyseistä ongelmaa omassa kotiverkkoyhteydessään.

Bufferbloat ja siihen liittyvät käsitteet ovat aiheena mielenkiintoinen, sekä ongelmana sellainen, johon kuka tahansa voi törmätä jakaessaan verkkoyhteyden muiden käyttäjien kanssa. Tietämällä ilmiötä ehkäisevistä ratkaisuista, niiden toiminnallisuudesta sekä ymmärtämällä aiheeseen liittyvät termien tarkoituksen, kuka tahansa pystyy ottamaan tämän tapaisen ohjelmistoratkaisun käyttöönsä ja kokeilemaan miten se vaikuttaa oman verkkoyhteyden laatuun. Viiveestä kärsivään verkkoyhteyteen ei ole ainoat ratkaisut ostaa uusi reititin tai päivittää verkkoyhteys nopeampaan, vaikkakin näillä ratkaisuilla voikin ratkaista tämän kyseisen ongelman tiedostamatta sitä.

LÄHTEET

Allman, M. 2012. Comments on bufferbloat. Viitattu 3.1.2023. <https://icir.org/mallman/papers/bufferbloat-ccr13.pdf>.

Bufferbloat.net. 2022. CoDel Overview. Viitattu 18.1.2023. <https://www.bufferbloat.net/projects/codel/wiki/>.

Bufferbloat.net. 2022. More about Bufferbloat. Viitattu 3.1.2023. https://www.bufferbloat.net/projects/bloat/wiki/More_about_Bufferbloat/.

Corbet, J. 2018. Let them run CAKE. Viitattu 22.1.2023. <https://lwn.net/Articles/758353/>.

dd-wrt.com. 2022. Hardware. Viitattu 07.1.2023. <https://wiki.dd-wrt.com/wiki/index.php/Hardware>.

Dell.com. 2022. PowerScale: iperf3-käyttö OneFS:ssä. Viitattu 23.1.2023. <https://www.dell.com/support/kbdoc/fi-fi/000188735/iperf3-k%C3%A4ytt%C3%B6-onefs-ss%C3%A4>.

Elisa.fi. 2022. Kodin nettiyhteydet. Viitattu 1.4.2023. <https://elisa.fi/kauppa/nettiliittymat/kiinteat-laajakaistat>.

Gettys, J. Bufferbloat: Dark Buffers in the Internet. Viitattu 6.1.2023. <https://ieeexplore.ieee.org/document/5755608>.

Høiland-Joergensen, T. McKeeney, P. Täht, D. Gettys, J. & Dumazet, E. 2018. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm. Viitattu 19.1.2023. <https://www.rfc-editor.org/rfc/rfc8290>.

Høiland-Jørgensen, T. Täht, D. & Morton, J. 2018. Piece of CAKE: A Comprehensive Queue Management Solution for Home Gateways, 1. Viitattu 20.1.2023. <https://arxiv.org/pdf/1804.07617.pdf>.

Jha, S. & Mahdub, H. 2002. Engineering Internet QoS. Viitattu 10.1.2023. https://www.researchgate.net/publication/220692303_Engineering_Internet_QoS.

kb.netgear.com. 2022. How QoS improves performance. Viitattu 9.1.2023. <https://kb.netgear.com/972/How-QoS-improves-performance>.

Nagle, J. 1985. On Packet Switches With Infinite Storage. Viitattu 2.1.2023. <https://www.ietf.org/rfc/rfc0970.txt>.

Nichols, K. & Jacobson, V. 2012. Controlling Queue Delay. Viitattu 4.1.2023. <https://queue.acm.org/detail.cfm?id=2209336>.

Nichols, K. Jacobson, V. McGregor, A. & Iyengar, J. 2018. Controlled Delay Active Queue Management. Viitattu 16.1.2023. <https://www.rfc-editor.org/rfc/rfc8289>.

OpenVault. 2022. Broadband Insights Report (OVBI) 1Q22. Viitattu 3.1.2023. <https://openvault.com/wp-content/uploads/2022/06/OVBI-1Q22-Report-FINAL.pdf>.

OpenWrt.org. 2021. NETGEAR R6220. Viitattu 8.1.2023. <https://openwrt.org/toh/netgear/r6220>.

OpenWrt.org. 2022. SQM Details. Viitattu 11.1.2023. <https://openwrt.org/docs/guide-user/network/traffic-shaping/sqm-details>.

Openwrt.org. 2022. SQM (Smart Queue Management). Viitattu 12.1.2023. <https://openwrt.org/docs/guide-user/network/traffic-shaping/sqm>.

Palmei, J. Gupta, S. Imputato, P. Morton, J. Tahiliani, M. P. Avallone, S. & Täht, D. 2019. Design and Evaluation of COBALT Queue Discipline. Viitattu 18.3.2023. <https://ieeexplore.ieee.org/abstract/document/8847054/authors#authors>.

Ubuntu Manpage Repository. 2022. CoDel. Viitattu 17.1.2023. <https://manpages.ubuntu.com/manpages/bionic/en/man8/tc-codel.8.html>

Waveform. 2022. Bufferbloat and Internet Speed Test. Viitattu 25.12.2022.
<https://www.waveform.com/tools/bufferbloat>