Aalto-yliopisto
Sähkötekniikan korkeakoulu

HTTP/3 suorituskyky

Kandidaatintyö

8. helmikuuta 2026

Saku Haataja

Tiivistelmä

HTTP/3 (Hypertext Transfer Protocol) on uusin versio HTTP-protokollasta ja se standardoitiin vuonna 2022. HTTP on sovelluskerroksen protokolla TCP/IP mallissa. HTTP/3 käyttää kuljetuskerroksen protokollanaan QUIC-protokollaa, joka on rakennettu UDP-protokollan päälle (User Datagram Protocol). QUIC-protokollaan on integroitu TLS 1.3-salausprotokolla (Transport Layer Security).

Tässä kandidaatintutkielmassa tutkitaan kirjallisuuskatsauksen keinoin HTTP/3:n suorituskykyä vertaamalla sitä aiempiin versioihin kolmella eri osa-alueella: verkkosivujen latausajoissa ja käyttökokemuksessa, mobiilikäytössä ja suoratoistossa. Tavoitteena on selvittää myös mistä suorituskykyedut johtuvat. Tutkimus on rajattu käsittelemään suorituskykyä kolmessa edellä mainitussa tilanteessa.

HTTP/3 on multipleksoitu protokolla eli se mahdollistaa useiden tietovirtojen (streams) siirtämisen yhden yhteyden yli. HTTP/3 eroaa HTTP/2:sta ratkaisemalla Head-of-line blocking ongelmaksi kutsutun tilanteen jossa saapumatta jäänyt paketti estää seuraavien pakettien lähetyksen. HTTP/3:n yhteydenmuodostus on myös edeltäjiään nopeampi lyhyemmän yhteydenmuodostuskättelyn ansiosta.

Verkkosivujen latausajoissa HTTP/3 on erityisen suorituskykyinen protokolla aiempiin versioihin verrattuna, kun yhteys on huono eli, kun viive on korkea tai kaistanleveys on heikko. Mobiilikäytössä aiempiin versioihin verrattuna HTTP/3 suoriutuu paremmin eri verkko-olosuhteissa.

Sisällys

1 Johdanto

HTTP/3 on uusin versio hypertekstin siirtoprotokollasta (Hypertext Transfer Protocol, HTTP). HTTP kehitettiin 1990-luvun alussa ja sittemmin siitä on tullut verkkoprotokollista käytetyin [1]. Keskeisenä erona aiempiin HTTP-versioihin on, että HTTP/3 käyttää kuljetusprotokollanaan UDP:n (User Datagram Protocol) päälle rakennettua multipleksoitua QUIC-protokollaa. Aiemmat HTTP-versiot käyttivät TCP-protokollaa (Transmission Control Protocol) kuljetuskerroksen protokollanaan.

Keskeisimpiä HTTP/3:n etuja on nopea yhteydenmuodostus, joka on mahdollista QUICin käytön vuoksi. HTTP/3 poistaa myös Head-of-line blocking -ongelman, joka oli ominaista TCP:tä käyttäville aiemmille versioille. Kolmantena etuna HTTP/3:ssa on tehokkaampi tiedon pakkaus ja priorisointi. Tämä on seurausta HTTP/3:n otsikkotietojen QPACK-pakkauksesta, joka on tehokkaampi kuin HTTP/2:n HPACK-pakkaus. Neljäs etu HTTP/3:ssa on sisäänrakennettu salaus ja tietoturva, jotka ovat seurausta QUIC-protokollan TLS 1.3 salausprotokollasta, joka tarjoaa vahvan tietoturvan protokollatasolla. Viides etu HTTP/3:ssa on aiempia versioita parempi suorituskyky heikon kaistanleveyden ja korkean viiveen olosuhteissa. Tämä on seurausta protokollan nopeammasta yhteyden muodostuksesta, paremmasta pakettien käsittelystä ja tehokkaammasta virheenkorjauksesta. QUIC-protokolla on suunniteltu siten, että se mahdollistaa helpon laajennettavuuden ja yhteensopivuuden tulevien teknologioiden kanssa.

Tässä kandidaatintutkielmassa syvennytään HTTP/3:n etuihin verrattuna aiempiin HTTP-versioihin. Tutkielma alkaa taustoittamisella, eli käytännössä tarkastelemalla protokollien historiaa ja kehitystä aiemmista versioista päätyen HTTP/3:een. Seuraavaksi esittellään, mikä HTTP/3 käytännössä on. Tutkielmassa esitellään myös HTTP/3:n kuljetusprotokolla QUIC sekä salausprotokolla TLS 1.3. Lopuksi syvennytään tutkimuksiin, joissa käsitellään HTTP/3:n suoriutumista kolmessa eri tilanteessa: verkkoselailussa, mo-

biilikäytössä sekä suoratoistopalveluiden käytössä. Keskeisimpänä tutkimuskysymyksenä, jota edellä esitetyt seikat tukevat, on HTTP/3:n suorituskyvyn edut verrattuna aiempiin protokollaversioihin. Tutkimus toteutetaan kirjallisuuskatsauksena.

2 Teoreettinen tausta

Tulevissa luvuissa käsitellään HTTP:n historiaa ja sen jälkeen esitellään protokollat: UDP ja TCP. HTTP:n aiempien versioiden tuntemus on olennaista, jotta voidaan verrata tämän tutkielman aiheena olevaa HTTP/3:a aiempiin versioihin. Protokolla TCP on toiminut aiempien HTTP-versioiden kuljetusprotokollana ja HTTP/3:n käyttämä QUIC-protokolla on rakennettu UDP-protokollan päälle. Siksi näiden protokollien käsittely on välttämätöntä, kun verrataan uusinta versiota aiempiin versioihin.

2.1 HTTP-protokollan historia

HTTP-protokolla on kehittynyt yli 30 vuoden historiansa aikana vastaamaan verkkoympäristön muuttuvia tarpeita. Ensimmäinen versio HTTP/0.9, joka tunnetaan myös nimellä HTTP, julkaisti vuonna 1990 Tim Berners-Lee. Kyseisellä versiolla oli mahdollista siirtää ainoastaan HTML-dokumentteja.[2]

Toinen versio protokollasta julkaistiin vuonna 1996. Tämä versio sai nimekseen HTTP/1.0. RFC 1945:n mukaan HTTP on perusrakenteeltaan yleiskäyttöinen, tilaton ja olio-orientoitunut protokolla. Se käyttää Uniform Resource Identifier (URI) -järjestelmää resurssien paikantamiseen ja tunnistamiseen. HTTP-viestit koostuvat pyyntö- ja vastausviesteistä, jotka sisältävät otsakkeita ja mahdollisen viestirungon. HTTP on luonteeltaan asymmetrinen eli HTTP-asiakas lähettää pyyntöviestin HTTP-palvelimelle ja palvelin vastaa siihen.[3]

HTTP/1.1 määriteltiin vuonna 1997 RFC 2068:ssa [4]. HTTP/1.1 sai päivityksen vuonna 1999, mikä määriteltiin RFC 2616:ssa [5]. HTTP/1.1:ssä dataa siirretään HTTP-viestien avulla. Viestejä on kahta tyyppiä: pyyntöjä

(requests) ja vastauksia (responses). Pyyntöviesteillä vastaanotetaan tai lähetetään dataa. Vastauksilla saadaan tulos pyyntöviestin toteumasta. Pyyntöviestit sisältävät pyyntörivin (Request Line) joka sisältää HTTP-metodin esim PUT tai GET, Request-URI:n ja HTTP-version. Vastausviestit koostuvat tilariveistä (Status-Line), jotka sisältävät HTTP-version, tilarivin esimerkiksi 2xx onnistunut tai 5xx palvelinvirhe ja syy-lausekkeen (Reason-Phrase), joka kuvailee tekstinä tilakoodia. HTTP-otsikot (headers) välittävät lisätietoja viestin mukana. [2]

Merkittävä parannus HTTP/1.0-versioon on välimuistiin tallentaminen (caching), joka tekee mahdolliseksi asiakkaiden (clients) aiempien vastausten tallentamisen vähentäen siten palvelimille tehtävien pyyntöjen määrää [2]. Parannuksia aiempaan versioon ovat myös pysyvät yhteydet, jotka mahdollistavat useiden pyyntöjen ja vastausten lähettämisen saman TCP-yhteyden kautta. Pysyvä yhteys vähentää toistuvia yhteydenmuodostuksia, joka taasen vähtentää ylikuormitusta, joka on seurausta toistuvista yhteydenmuodostuksista. HTTP/1.1 sisältää myös pipelining-ominaisuuden, joka mahdollistaa useiden pyyntöjen lähettämisen ennen aiempien pyyntöjen vastausten vastaanottamista. [2]

HTTP/2 standardoitiin vuonna 2015 standardissa RFC 7540 [6]. HTTP/2 mahdollistaa viiveen vähentämisen ja verkon resurssien tehokkaamman käytön, otsikkokenttien pakkauksen (header field compression) avulla ja sallimalla useat samanaikaiset vaihdot saman yhteyden kautta. HTTP/2:n ominaisuus on, että palvelin voi lähettää dataa asiakkaalle (client) ilman erillistä pyyntöä. [6]

HTTP/2 on rakennettu HTTP/1.1:n ydinsemantiikan päälle. HTTP/1.1:stä poiketen HTTP/2 on binääriprotokolla. Binääriprotokolla tarkoittaa, että viestirunko voidaan lähettää binäärimuodossa. Binäärimuodon ansiosta viestit pakkautuvat pienempään tilaan. Otsikkotietojen pakkaaminen HPACK-mekanismilla vähentää siirrettävän datan määrää. Uutta HTTP/2:ssa on myös multipleksointi. Multipleksoinnilla tarkoitetaan, että viestintä palvelimen ja asiakkaan välillä tapahtuu yhden TCP-yhteyden kautta. Tämä yhteys jaetaan useisiin virtauksiin (streams). Jokainen pyyntö ja vastaus saa oman virtansa. Sekä asiakkaat, että palvelimet voivat avata omat virtauksensa. [2]

Vuonna 2022 julkaistiin standardi RFC 9114, jossa määritettiin uusin versio HTTP/3 [7]. Aiemmissa versioissa käytetyn TCP-protokollan sijaan HTTP/3 käyttää QUIC-protokollaa. HTTP/3:n etuja ovat muun muassa nopea yhteyden muodostus, multipleksointi ja parannettu tiedon pakkaus käyttäen

QPACK otsikoiden pakkausmenetelmää.[1]

2.2 TCP-protokolla

Tässä luvussa esitellään TCP-protokolla (Transmission Control Protocol). TCP-protokollaa käytetään HTTP/3:n edeltäjissä kuljetuskerroksen protokollana. Tässä luvussa esitellään myös lyhyesti ongelmia, joita TCP-protokollaan liittyy. Kyseiset ongelmat liittyvät olennaisesti siihen, miksi QUIC-protokolla on kehitetty.

TCP-protokolla standardoitiin vuonna 1974 RFC 675-standardissa. Uusin versio TCP-protokollan standardista on RFC 9293. TCP on kuljetuskerroksen protokolla internetin protokollapinossa. TCP:n keskeinen pyrkimys on varmistaa luotettava tiedonsiirto, siten että lähetetty data saapuisi vastaanottajalle oikeassa järjestyksessä ja ilman virheitä. Vastaanotettujen tietojen oikea järjestys pyritään varmistamaan sekvenssinumeroilla. Kukin datajakso sisältää tarkastussumman, jolla varmistetaan datan vioittumattomuus lähetyksen aikana. Jos data on vioittunut lähetetään se uudelleen. [8]

TCP on yhteysorientoitunut protokolla (connection-oriented). Tämä tarkoittaa, että ennen datan siirtämistä on luotava lähettäjän ja vastaanottajan välille kolmivaiheisen kättelyn (three-way-handshake) avulla. Näin varmistetaan, että osapuolet ovat valmiita tiedonsiirtoon. TCP tukee myös kaksisuuntaista tiedonsiirtoa.[8]

TCP:n tapa varmistaa pakettien järjestys aiheuttaa Head-of-line blocking ongelmaksi kutsutun tilanteen, mikä tarkoittaa, että jos paketti on kadonnut tai viivästynyt seuraavat paketit joutuvat odottamaan kunnes kadonnut tai viivästynyt paketti lähetetään uudelleen ja vastaanotetaan onnistuneesti. Tämä hidastaa tiedonvälitystä eli johtaa lisääntyneeseen viiveeseen. [2]

TCP sisältää kuitenkin useita ruuhkanhallinta algoritmeja kuten hidas aloitusalgoritmin (slow start) ja ruuhkan välttely-algoritmin (Congestion avoidance), jotka on määritelty RFC:ssä 2581. Pakettien katoaminen on yleensä seurausta ruuhkasta internetissä.[8] Ruuhkan aiheuttamiin häiriöihin TCP-protokolla pyrkii vastaamaan muun muassa edellä mainituilla algoritmeilla.

Toinen TCP:hen liittyvä haaste on kolmivaiheisesta kättelystä johtuva viive. Kuten aiemmin kuvattiin, TCP-protokolla edellyttää kolmivaiheista kättelyä

ennen tiedonsiirtoa. Kolmivaiheinen kättely aiheuttaa siten myöskin osaltaan viivettä. [2]

2.3 UDP-protokolla

Tässä luvussa esitellään lyhyesti UDP-protokolla ja käsitellään sen eroja TCP-protokollaan. QUIC-protokolla, jota käsitellään luvussa 3.1 tarkemmin on HTTP/3:ssa käytettävä protokolla, joka on rakennettu UDP-protokollan päälle.

UDP (User Datagram Protocol) on kevyt kuljetuskerroksen protokolla. Se standardoitiin vuonna 1980 RFC 768:ssa [9]. UDP ei sisällä ruuhkanhallintamekanismeja TCP:n tapaan. UDP ei myöskään lähetä paketteja uudelleen kuten TCP tekee. UDP on yhteydetön protokolla, mikä tarkoittaa, että sen avulla voidaan lähettää yksittäisiä paketteja (datagram) ilman kättelyitä tai erillistä yhteyden muodostamista. [9] Edellä mainitut seikat tekevät UDP:stä kevyen protokollan, mikä nopeuttaa sen toimintaa, kun verrataan TCP:n kolmivaiheisen kättelyn aiheuttamaan viiveeseen.

Eroja TCP:n ja UDP:n välillä on useita. TCP varmistaa pakettien saapumisen perille ja, jolleivat ne ole saapuneet lähettää ne uudelleen. UDP ei tee kumpaakaan. TCP lisää paketteihin järjestysnumerot ja varmistaa, että paketit saapuvat oikeassa järjestyksessä. UDP ei välitä pakettien saapumisjärjestyksestä. TCP on ominaisuuksiensa vuoksi hitaampi kuin UDP. TCP vaatii enemmän tietokoneen resursseja huomioidessaan pakettien häviämisesen ja niiden järjestyksen.[10]

3 HTTP/3

Seuraavissa luvuissa käsitellään HTTP/3:n arkkitehtuuria ja esitellään sen etuja verrattuna aiempiin HTTP-versioihin. Luvuissa esitellään myös QUIC-protokolla, joka toimii HTTP/3:n kuljetuskerroksen protokollana. QUIC-protokollaan on integroitu salausprotokolla TLS 1.3, joka myös esitellään. Lopuksi käsitellään keskeisiä etuja, joita HTTP/3:ssa on sekä HTTP/3:n etujen syitä.

3.1 QUIC-protokolla

Tässä luvussa esitellään HTTP/3:n käyttämä QUIC-protokolla. QUIC-protokolla on luvussa 2.3 esitellyn UDP-protokollan päälle rakennettu multipleksoitu siirtoprotokolla.

Googlen Jim Roskindin kehittämän QUIC-protokollan ensimmäinen versio julkaistiin vuonna 2012. Aluksi nimi QUIC viittasi lyhenteeseen sanoista Quick UDP Internet Connections, kunnes IETF standardoi protokollan standardissa RFC 9000, jossa QUIC-protokollan ei katsottu olevan lyhenne, vaan olevan ainoastaan protokollan nimi.[11]

QUIC-protokolla voi ohittaa TCP-protokollan hidastavia vaiheita, kuten kolmivaiheisen kättelyn yhteydenmuodostuksessa. QUIC yhteydenmuodostuksessa voidaan käyttää 0-RTT:ta (zero round-trip time), jossa asiakas (client) lähettää dataa samalla, kun yhteys muodostetaan. Vaihtoehtoisesti voidaan käyttää 1-RTT (One round-trip time) yhteydenmuodostusta, jossa lähetetään asiakas avaa yhteyden salauksen sisältävällä aloitusviestillä, johon palvelin vastaa salausviestin sisältävällä vastausviestillä.[11]

QUIC-protokolla mahdollistaa useiden tietovirtojen (streams) lähettämisen

ja vastaanottamisen samanaikaisesti. Tätä kutsutaan multipleksoinniksi.[11] Myös HTTP/2 on multipleksoitu protokolla.[6] QUICin multipleksointi eroaa HTTP/2:sta siten, että multipleksointi suoritetaan HTTP/3:ssa kuljetusprotokollan tasolla eli QUIC-protokollan sisällä eikä HTTP-tasolla kuten HTTP/2:ssa.[11]

TCP-protokollaa käsittelevässä luvussa 2.2 esiteltiin kyseiseen protokollaan liittyvä Head-of-line blocking ongelma, joka tarkoitti, että kadonnut tai väärässä järjestyksessä saapunut paketti keskeyttää datan siirron, kunnes paketti saadaan perille. QUIC-protokollan multipleksointi ratkaisee tämän ongelman jatkamalla datan siirtoa muita tietovirtoja pitkin vaikka yksi tietovirta tukkeutuisi paketin kadottua[1]. Tämä sulavoittaa protokollan toimintaa ja vähentää viivettä, joka aiheutuisi kadonneista paketeista.

QUIC-protokollaan on integroitu TLS 1.3 -salausprotokolla, joka tarjoaa vahvan suojan tiedonsiirron aikana. Kaikki tietoliikenne on oletusarvoisesti salattu, mikä parantaa protokollan käyttäjien tietoturvaa ja yksityisyyttä. [11]

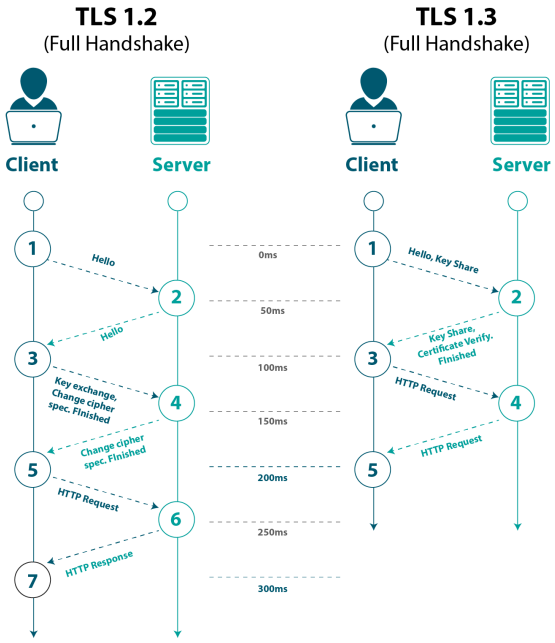
3.2 TLS 1.3 -protokolla

HTTP/3 käyttää kuljetuskerroksen protokollanaan QUIC-protokollaa. QUIC-protokollaan on integroitu osaksi TLS 1.3 -tietoturvaprotokolla (transport layer security). Tässä luvussa esitellään TLS 1.3:n toimintaa.

IETF julkaisi TLS 1.3:a käsittelevän standardin RFC 8446 vuonna 2018[12]. TLS 1.3 on tietoturvaltaan paranneltu versio edeltäjistään. Se ei tue vanhentuneita salausalgoritmeja kuten RC4:ää, vaan käyttää moderneja salausalgoritmeja AET-CGM:ää ja ChaCha20-Poly1305:a[12].

TLS 1.3 suorituskyky on parempi kuin aiempien versioiden. TLS 1.3 vaatii vain yhden kättelykierroksen 1-RTT (one round-trip) muodostaakseen yhteyden asiakkaan (client) ja palvelimen (server) välille. ja se tukee myös 0-RTT:ä (zero round-trip). Tämä pienentää ylimääräisistä kättelykierroksista aiheutuvaa viivettä. [13] Kuvassa 3.1 on kuvattu erot TLS 1.2 ja TLS 1.3 kättelyiden välillä. TLS 1.2 kättelyssä asiakas (client) ja palvelin (server) tervehtivät ensin toisiaan (client hello ja server hello), jonka jälkeen ne vaihtavat avaimia keskenään (key exchange). Edellä mainituista tapahtumista koostuu TLS 1.2:n kaksivaiheinen kättely. TLS 1.3:ssa asiakkaan ja palvelimen välinen tervehdyskierros ja avainten vaihto tapahtuu samanaikaisesti eli sitä kutsutaan

1-RTT kättelyksi.



Kuva 3.1: TLS 1.2 ja TLS 1.3 kättelyt. [14].

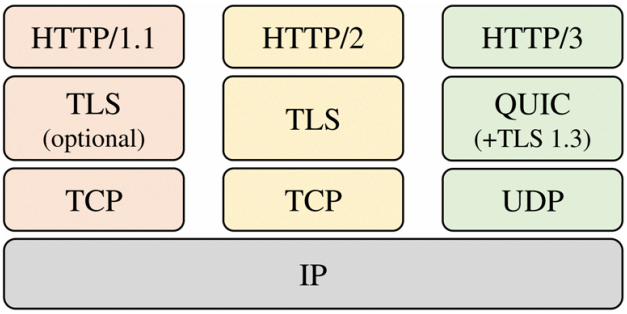
TLS 1.3 mahdollistaa myös 0-RTT kättelyn. 0-RTT on mahdollista jos asiakas on aiemmin ollyt yhteydessä palvelimeen ja saanut ennalta jaetun avaimen (pre-shared key, PSK). Kun asiakas seuraavan kerran ottaa yhteyttä palvelimeen, se voi käyttää PSK:tä ja lähettää dataa jo ensimmäisellä kättelykierroksella.[12] 0-RTT yhteydenmuodostus sisältää kuitenkin riskin toistohyökkäykselle (replay attack) altistumiselle. [11] Toistohyökkäyksellä tarkoitetaan tilannetta, jossa hyökkääjä sieppaa ja toistaa viestin tai datan. [13]

3.3 HTTP/3 arkkitehtuuri ja toiminta

Tässä luvussa keskitytään käsittelemään HTTP/3-protokollan arkkitehtuuria ja toimintaa. Luvussa 2.1 esiteltiin jo pääpiirteet aiemmista HTTP-protokollan versioista. Luvussa 2.3 Esiteltiin UDP-protokolla ja luvussa 3.1 QUIC-protokolla, joita HTTP/3-protokollassa käytetään. Kandidaatintutkielman tulevissa lu-

vuissa tullaan taasen syventymään siihen, mitä suorituskyvyllisiä etuja HTTP/3-protokollassa on verrattuna aiempiin versioihin.

HTTP/3 on uusin versio HTTP-protokollasta. Se on suunniteltu parantamaan suorituskykyä ja ratkaisemaan aiempien versioiden haasteita kuten HTTP/2-protokollan Head-of-line blocking ongelman.[2] Muita keskeisiä parannuksia HTTP/2:een verrattuna ovat TCP:n korvaaminen QUIC-protokollalla, tehokkaampi otsikoiden (headers) pakkaaminen ja parannetut turvallisuusominaisuudet pohjautuen TLS 1.3:n käyttöön.[1]



Kuva 3.2: HTTP-protokollapinot ja niiden kerrokset. [1].

Kuvassa 3.2 on esitelty HTTP-protokollapinojen rakenne. HTTP/3 Käyttää kuljetuskerroksen protokollana QUIC-protokollaa, joka on esitelty tarkemmin luvussa 3.1. QUIC-protokollaan on integroitu TLS 1.3-salausprotokolla, jota käsiteltiin luvussa 3.2. QUIC-protokolla toimii UDP-protokollan päällä, joka esiteltiin luvussa 2.3.

HTTP/3 protokollan toimintaa nopeuttaa QUIC-protokollan yhteydenmuodostus, jossa HTTP/2:n kolmivaiheisen kättelyn sijaaan voidaan jo yhteydenmuodostusvaiheessa samanaikaisesti jakaa TLS 1.3 salaukset[11]. HTTP/3 toimintaan vaikuttaa tehostavasti myös multipleksointi, jonka avulla QUIC-protokolla muodostaa useita tietovirtoja (streams), joissa dataa voidaan yhteydenmuodostuksen jälkeen siirtää samanaikaisesti[15]. Multipleksointi ja UDP:n päälle rakennettu QUIC-protokolla ratkaiseavt myös HTTP/2:een liittyvän keskeisen haasteen Head-of-line blocking -ongelman, joka tarkoittaa, että saapumaton paketti estää seuraavien pakettien lähetyksen kunnes puuttuva paketti saadaan lähetettyä onnistuneesti. Multipleksoidussa HTTP/3 protokollassa muut virrat voivat jatkaa tiedonsiirtoa, vaikka jokin paketti puuttuisikin. [2]

3.4 HTTP/3:n ominaisuudet ja edut

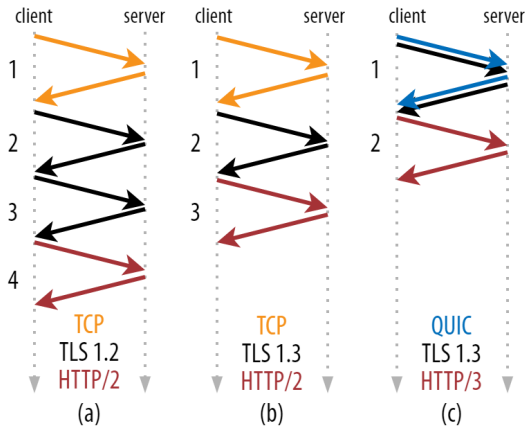
Seuraavissa alaluvuissa käydään läpi keskeisimpiä HTTP/3:n etuja verrattuna aiempiin HTTP-versioihin. Luvussa 2.1 käytiin läpi protokollan aiempien versioiden ominaisuuksia ja historiaa. Vertaamalla aiempiin versioihin voidaan huomata kuinka HTTP/3 on kehittynyt edeltäjäversioistaan. Luvussa 4 paneudutaan siihen, millaisia vaikutukset ovat internetin käyttäjille.

3.4.1 Nopea yhteyden muodostaminen ja viiveen vähentäminen

HTTP/3: viive on pienempi kuin aiempien HTTP-versioiden, erityisesti tilanteissa, joissa viive on korkea tai kaistanleveys on heikko[1]. Ensimmäinen seikka, joka vähentää HTTP/3:n viivettä on QUIC-protokollan tapa muodostaa yhteys 0-RTT kättelyllä tai 1-RTT kättelyllä. HTTP/2:ssa käytettiin kolmivaiheista kättelyä, joka hidastaa yhteydenmuodostusta verrattuna QUIC-protokollan kättelyyn.[2] Kuvassa 3.3 on kuvattu erot HTTP/2:n käyttämän TCP-kuljetusprotokollan yhteydenmuodostuksesta käytettäessä TLS-protokollan versioita 1.2 ja 1.3 sekä HTTP/3:n kuljetusprotokollan QUIC:n yhteydenmuodostuksesta. HTTP/2:sen kättelyssä suoritetaan ensin TCP-kättely (keltaiset nuolet kuvassa), jonka jälkeen TLS-kättelykierrokset (mustat viivat kuvassa), joita on kaksi käytettäessä TLS 1.2:ta tai yksi käytettäessä TLS 1.3:a. Näiden jälkeen voidaan aloittaa datan siirtäminen (punaiset viivat). Kuvassa on HTTP/3:n osalta kuvattu 1-RTT-kättely, jossa QUIC-kättely (siniset nuolet kuvassa) ja TLS-kättely suoritetaan samalla kierroksella. Näiden jälkeen on mahdollista aloittaa datan siirtäminen. Kuva havainnollistaa, että QUIC-protokollan kättely vaatii vähemmän kierroksia kuin edeltävät versiot.

3.4.2 Multipleksointi ja Head-of-line blocking -ongelman ratkaisu

HTTP/3:n käyttämä QUIC-protokolla mahdollistaa useiden tietovirtojen (streams) siirtämisen yhden yhteyden yli[11]. Myös versiossa HTTP/2 on mahdollista lähettää tietovirtoja multipleksoidusti, mutta erona HTTP/3:een HTTP/2:n multipleksointi tapahtuu sovelluskerroksen tasolla eli HTTP tasolla, kun HTTP/3:ssa



Kuva 3.3: Yhteydenmuodostus TCP vs QUIC [16].

multipleksointi tapahtuu alemmalla kerroksella kuljetusprotokollatasolla QUIC-protokollan sisällä[6][11]. Multipleksointi mahdollistaa useiden pyyntöjen ja vastausten lähettämisen samanaikaisesti, joka taasen mahdollistaa tehokkaamman kaistanleveyden käytön verkkoyhteydelle[2].

HTTP/2:n multipleksoinnin haaste on Head-of-line blocking -ongelma jossa yksi kadonnut tai viivästynyt paketti estää kaikkien muiden saman yhteyden kautta lähetettävien tietovirtojen etenemisen, kun TCP yrittää uudelleenlähettää pakettia. Tämä on seurausta HTTP/2:n kuljetusprotokollana toimivan TCP:n palvelusemantiikasta, jossa tavut toimitetaan ylemmälle protokollakerrokselle eli HTTP:lle siinä järjetyksessä, jossa ne on lähetetty.[17] HTTP/3:n multipleksoitu yhteys ei keskeytä tiedonsiirtoa, kun odotetaan kadonnutta pakettia, vaan jatkaa tiedonsiirtoa kadonneesta paketista huolimatta muiden tietovirtojen kautta. HTTP/3 ei odota HTTP/2:n tavoin pakettien saapuvan samassa järjestyksessä ja siten se ratkaisee Head-of-line blocking ongelman. Tämä on seurausta QUIC-protokollan ja TCP-protokollan eroista koska UDP:n päälle rakennettu QUIC ei edellytä pakettien saapumista oikeassa järjestyksessä tietovirtojen välillä.[2]

3.4.3 Sisäänrakennettu salaus ja tietoturvaominaisuudet

HTTP/3:n kuljetuskerroksen protokolla QUIC:iin on integroitu TLS 1.3-salausprotokolla[2]. TLS 1.3 takaa vahvan salauksen ja tietoturvan kaikel-

le liikenteelle. Kaikki asiakkaan (client) ja palvelimen (server) välinen kommunikaatio on salattua, mikä suojaa sekä asiakkaita että palvelinta verkkovakoilulta. QUIC-protokollassa myös otsikkotiedot (headers) ovat yleensä salattuja, mikä lisää käyttäjien yksityisyyttä.[18]

HTTP/3 tarjoaa paremman suojan palvelunestohyökkäyksille (Denial of Service, DoS) käyttämällä QUIC-protokollaa, joka salaa yhteydenmuodostuksen. Yhteydenmuodostuksen salaus tarjoaa suojaa esimerkiksi vahvistushyökkäyksiä (amplification attack) vastaan. [18]

Merkittävä parannus tietoturvaan on myös HTTP/3:n yhteyden siirto (connection migration). Yhteyden siirto vähentää aikaa, jonka yhteys on alttiina yhteyden kaappaamiselle (connection hijacking).[18] QUIC-protokollassa yhteyden siirto on mahdollista connection ID:n ansiosta. Vaikka päätelaitteen IP-osoite muuttuisi esimerkiksi siirryttäessä kännykällä wifistä mobiiliverkkoon, voidaan yhteys säilyttää, kun palvelin ja asiakas muistavat aiemman connection-ID:n. Yhteyden siirto on mahdollista vain kun yhteydenmuodostus kättely on suoritettu loppuun.[11]

4 Vaikutukset webliikenteeseen

Seuraavissa luvuissa tutkitaan tutkimuskirjallisuutta liittyen HTTP/3:n suorituskykyyn. Ensimmäisenä tutkitaan verkkosivujen latausaikoja ja käyttäjäkokemusta, joita HTTP/3 on pyrkinyt parantamaan muun muassa nopeammalla yhteydenmuodostuksella. Toiseksi tarkastellaan HTTP/3:n suorituskykyä mobiililaitteilla. Muun muassa mahdollisuus yhteyden siirtoon (connection migration) ja nopea yhteydenmuodostus tekijöitä, joiden oletetaan parantavan HTTP/3:n suorituskykyä mobiiliverkoissa. Kolmantena tarkastellaan HTTP/3:n suorituskykyä suoratoistossa, joka on todella suosittua sekä video että audio sisällöissä. Suorituskyky tutkimukset, joihin tässä tutkielmassa on paneuduttu keskittyvät pääasiassa vertailemaan HTTP/3:sta versioihin: HTTP/2 ja HTTP/1.1.

4.1 Verkkosivujen latausajat ja käyttäjäkokemus

Verkkosivujen latausajat ovat kriittinen tekijä verkkosivustojen käyttäjien käyttäjäkokemuksen kannalta. Hitaasti latautuvat verkkosivut tekevät esimerkiksi lehden lukemisesta tai verkko-ostosten tekemisestä epämiellyttävämmän kokemuksen käyttäjälle.

Trevisan et al. kirjoittaman artikkelin mukaan HTTP/3 tarjoaa suorituskykyetuja erityisesti huonoissa verkko-olosuhteissa. Heidän mukaansa korkean viiveen tai heikon kaistanleveyden tilanteissa HTTP/3 menestyi muita versioita paremmin. Trevisan et al. vertailivat 14707 verkkosivua käyttäen HTTP/2:ta ja HTTP/3:a. Korkean viiveen tilanteissa 32% verkkosivuista olivat nopeampia käytettäessä HTTP/3:a, 11% oli nopeampia käytettäessä HTTP/2:ta ja 57% olivat yhtä nopeita kummallakin versiolla. Eniten

HTTP/3:sta hyötyivät verkkosivut, jotka olivat rajoitetun määärän yhteyksiä ja kolmansien osapuolien palvelimia omaavia. Tapauksissa, joissa oli korkea pakettien katoamisaste HTTP/3 ja HTTP/2 suoriutuivat kutakuinkin samalla tavalla. Myös tilanteissa, joissa verkkoyhteys oli hyvä ei eroa HTTP/3:n ja HTTP/2:n välillä ollut havaittavissa.[1]

Guptan ja Bartosin kirjoittamassa artikkelissa tarkastellaan HTTP/3 suorituskykyä QOE-mittariston (quality of experience) avulla. Käytännössä artikkelissa arvioidaan QOE:ta läpäisykyvyn (throughput) ja FCP:n (first contentful paint) avulla. Läpäisykyky mittaa kuinka paljon dataa siirretään tietyn ajan kuluessa verkkoyhteyden kautta.[19] FCP on Googlen Lighthouse mittaristoon kuuluva mittari. Google lighthouse on verkkosivujen suorituskyvyn mittaus työkalu. FCP mittaa aikaa, joka sivustolta kuluu ladata ensimmäinen kuva käyttäjän navigoitua sivulle. [20] Artikkelin mukaan HTTP/3 suoriutui HTTP/2:a paremmin globaaleissa langattomissa verkoissa, kun taas HTTP/2 menestyi paremmin paikallisverkoissa. Selittäväksi seikaksi HTTP/3:n suoriutumiselle haastavissa olosuhteissa artikkelissa esitetään QUIC-protokollan multipleksointi ja sen kyky ratkaista Head-of-line blocking ongelma.[19]

Artikkelissa "Comparing Communication Efficiency: HTTP/3 versus HTTP/1.1 Latency in RESTful APIs"vertaillaan HTTP/3:a HTTP/1.1:en RESTful API -kehityksessä (Representational State Transfer application programming interface). Tutkimuksen mukaan HTTP/3 tarjoaa huomattavia parannuksia latausajoissa verrattuna HTTP/1.1:een. Pääasiallinen syy on HTTP/3:n QUIC-protokolla, nopeamman yhteydenmuodostuksen ansiosta vasteaika (response time) oli 0.236 sekuntia, kun HTTP/1.1:n vasteaika oli 0.373 sekuntia. Myös latausnopeudessa HTTP/3 oli merkittävästi nopeampi kuin HTTP/1.1. HTTP/3:n latausnopeus oli keskimäärin 229567 bittä sekunnissa, kun HTTP/1.1:n latausnopeus oli 144891 bittiä sekunnissa.[21] Vaikka tutkimus ei suoraan kerrokaan verkkosivujen latausnopeuksista, kertoo se silti HTTP/3:n tehokkuudesta ja antaa osviittaa sen suoriutumisesta HTTP/1.1:een verrattuna.

HTTP/3 ei suorituskykytutkimusten mukaan ole kaikissa tilanteissa ylivertainen protokolla. Verkkosivujen latausnopeudessa ja käyttäjäkokemuksessa se kuitenkin ylittää edeltäjä versionsa suoriutumisen erityisesti, kun yhteys on huono eli kaistanleveys on heikko tai viive on korkea. Tutkimuksia, joissa vertaillaan suoraan eri HTTP versioiden nopeuksia ei ole kovin useita, ja tässä tutkielmassa pyrittiin esittämään niistä edustava otos.

4.2 Mobiililaitteiden suorituskyky

Artikkelissa "A first look at HTTP/3 adoption and performance"tutkittiin HTTP/3:n ja HTTP/2:n suorituskykyä verkkoselailussa tabletilla ja älypuhelimella 3G -ja 4G -verkoissa. HTTP/3:n ja HTTP/2:n suoriutumista vertailtiin kahdella mittarilla: OnLoad tarkoittaa hetkeä, jolloin kaikki verkkosivun elementit on ladattu ja ovat valmiina käyettäväksi. SpeedIndex on mittari, joka mittaa, kuinka nopeasti verkkosivun näkyvät osat tulevat näkyviin latauksen aikana. OnLoad mittarilla HTTP/3 menestyi HTTP/2:ta paremmin kaikissa tilanteissa, käytettäessä mobiililaitteita. Suurimmat hyödyt HTTP/3 käytöstä saatiin 3G verkon tapauksessa älypuhelimelle suunnitelluilla verkkosivuilla ja huonoilla verkkoyhteydellä. Yhtä suuri hyöty saatiin käytettäessä tietokoneversiota verkkosivusta ja hyvällä yhteydellä. Myös SpeedIndex mittarilla HTTP/3 suoriutui HTTP/2:ta paremmin kaikissa testiasetelmissa. SpeedIndexillä mitattuna HTTP/3 hyötyi eniten hyvästä verkosta ja tietokoneelle versiosta verkkosivuilla verrattuna HTTP/2:een. [22]

Edellä kuvattu tutkimus tehtiin käyttäen Google Chrome selainta[22]. Tulokset voisivat olla erilaisia käytettäessä muita selaimia. Tutkimuksessa ei myöskään vielä ollut mukana 5G verkkoja, jotka myös osaltaan kaipaisivat suorituskykymittauksia. Yleisesti tutkimus osoitti saman kuin luvun 4.1 tutkimukset sen osalta, että HTTP/3 suoriutuu paremmin verrattuna HTTP/2:een ja HTTP/1.1:een, kun verkon viive on korkea[22]. Mobiiliverkon viive ja kaistanleveys vaihtelevat suuresti, kun siirrytään etäämmälle tukiasemasta ja yhteyteen käytettävä tukiasema vaihtuu. Siksi tulokset HTTP/3:n hyvästä suoriutumisesta mobiilissa eivät ole yllättäviä.

4.3 Suoratoistopalveluiden tehokkuus

Edellisen luvun tutkimuksessa tutkittiin edellisessä luvussa käsiteltyjen asioiden lisäksi myös HTTP/3:n HTTP/2:n ja HTTP/1.1:n suorituskykyä videoiden suoratoistamisessa (video streaming). Tutkimusta oli tehty erilaisissa verkko-olosuhteissa. Mittaristona tutkimuksessa käytettiin videon resoluutiota, PSD:tä (Playback Startup Delay), jolla tarkoitetaan aikaa joka kuluu videon käynnistymisen käyttäjän pyynnöstä ja videoiden laadun alaspäin pudottamista (Frequency of video downscale). Hyvissä verkko-olosuhteissa kaikki HTTP-versiot menestyivät samalla tavalla. Kun tutkittiin, miten eri ver-

siot reagoivat kontrolloituun pakettien häviämiseen (controlled packet loss), HTTP/2 ja HTTP/1.1 pudotti kuvanlaatua toisinaan. HTTP/3 osoittautui niitä vakaammaksi kyseisessä tilanteesa, sillä se ei pudottanut kuvanlaatuaan. Matalalla kaistanleveydellä 1 Mbit/s jokainen HTTP versioista pudotti videon laatua tasaisesti. 2 Mbit/s kaistanleveydellä ainoastaan HTTP/3 pudotti videon laatua ja siten menestyi heikoimmin. 5 Mbit/s kaistanleveydellä kaikki versiot menestyivät tasaisen hyvin. Rajallisen kaistanleveyden tilanteessa HTTP/3 myös menestyi huonoimmin PSD-mittarilla. Tutkimuksessa ei oltu varmoja, mistä HTTP/3:n heikko menestys rajallisen kaistanleveyden tilanteessa johtuu. Tutkijat esittävät hypoteesin, että tämä johtuu QUICin ruuhkanhallinta-algoritmien (congestion control algorithm) eroista TCP:n päälle rakennettuihin HTTP/2:een ja HTTP/1.1:een. Asiakkaan videon toistin vaikutti tutkimuksen mukaan aggressiivisemmalta käytettäessä HTTP/3:sta pyrkien lähettämään samaa videon osaa (chunk) eri resoluutioilla. Tämän tutkimuksessa arvellaan johtuvan viiveistä alemmilla verkkokerroksilla. [22]

Chellappan ja Bartosin kirjoittamassa tutkimuksessa: "Is QUIC Quicker with HTTP/3? An Empirical Analysis of Quality of Experience with DASH Video Streaming"testattiin HTTP/3:n suoriutumista videoiden suoratoistosta erilaisilla Adaptive Bit Rate (ABR) algoritmeilla vaihtelevilla verkko-olosuhteilla. ABR:t ovat suoratoistossa käytettävä algoritmeja, joilla mukautetaan automaattisesti videon laatua (bitrate) käyttäjän internetyhteyden mukaan. ABR tavoite on parantaa käyttäjän katselukokemusta (Quality of experience, QoE), vähentämällä puskurointia ja minimoimalla videon laadunvaihtelua. Tutkimuksen mukaan HTTP/3 saavutti HTTP/2:ta paremman QoE:n tilanteessa, jossa kaistanleveys heikkeni (BW-loss scenario). HTTP/3 myös avasi videon HTTP/2:ta nopeammin. HTTP/3:lle parhaiten soveltuvia ABR algoritmeja olivat matalan viiveen algoritmit (low-latency algorithms), joita ovat LearntoAdapt (L2A) ja Low-on-latency(LoL +). Niillä HTTP/3 saavutti korkeamman kuvan laadun, vähemmän laadunvaihteluita, paremman läpimenon (throughput) ja vähemmän bufferointia. [23]

5 Yhteenveto ja johtopäätökset

Tässä tutkielmassa perehdyttiin HTTP/3:n suorituskykyyn ja sen etuihin verrattuna aiempiin HTTP-versioihin erityisesti HTTP/1.1:een ja HTTP/2:een. Työn alussa käsiteltiin HTTP-protokollan historiaa ja kehitystä, joka mahdollistaa ymmärryksen, mikä protokollassa on muuttunut vuosien saatossa ja esiteltiin HTTP/2:n kuljetuskerroksen protokolla TCP. HTTP/3 käyttää kuljetuskerroksen protokollana QUICIa, joka esiteltiin luvussa 3.1. QUIC on rakennettu UDP-protokollan päälle ja UDP-protokolla esiteltiin luvussa 2.3. QUIC-protokollaan on integroitu TLS 1.3-salausprotokolla, joka esiteltiin luvussa 3.2.

Tutkielmassa käsiteltyjen standardien ja artikkelien mukaan HTTP/3:n keskeinen etu on nopeampi yhteydenmuodostus, joka on seurausta TLS 1.3-salausprotokollan käytöstä, jossa kättely kierroksia on vain yksi, sekä QUIC-protokollan kättelystä, joka voidaan suorittaa samaan aikaan TLS-kättelyn kanssa. HTTP/3 mahdollistaa myös 0-RTT yhteydenmuodostus. Toinen keskeinen etu HTTP/3:ssa on Head-of-line blocking ongelman ratkaisu, joka oli ominaista HTTP/2:lle. Kolmas keskeinen etu HTTP/3:ssa on multipleksointi, joka on toteutettu kuljetusprotokolla QUICin tasolla mahdollistaen usean tietovirran (stream) siirtämisen yhden yhteyden kautta. Neljäs etu on paranneltu tietoturva, joka on seurausta TLS 1.3-salausprotokollan käytöstä. TLS 1.3 tarjoaa suojatun tiedonsiirron koko yhteyden ajalle yhteydenmuodostuksesta alkaen.

Tutkielmassa selvitettiin HTTP/3:n suorituskykyä kolmella eri osa-alueella: verkkosivujen latausnopeuksissa ja käyttäjäkokemuksen kannalta, mobiilikäytössä ja suoratoistopalveluissa. HTTP/3:a verrattiin tutkielmaan käytetyssä kirjallisuudessa protokollaversioihin HTTP/1.1 ja HTTP/2. Verkkosivujen latausnopeuksissa ja käyttäjäkokemuksen kohdalla HTTP/3 on erityisen suorituskykyinen verrattuna HTTP/2:een kun yhteys on huono eli kaistanleveys on heikko ja latenssi on korkea. Muissa tilanteissa HTTP/3

suoriutui vähintään yhtä hyvin kuin aiemmat protokollaversiot. Mobiilikäytössä HTTP/3 oli tutkimusten mukaan erittäin suorituskykyinen protokolla päihittäen aiemmat versiot kaikissa olosuhteissa. Nämä suorituskyky edut ovat seurausta nopeammasta yhteydenmuodostuksesta, multipleksoinnista ja HOL-ongelman ratkaisusta. Suoratoistossa suorituskyky tutkimukset antoivat hieman ristiriitaisia tuloksia, joten niissä lisätutkimukset olisivat varmasti tarpeen. Yhteenvetona voidaan todeta, että HTTP/3 on erityisen suorituskykyinen mobiilikäytössä ja huonojen verkko-olosuhteiden tilanteissa.

HTTP/3:een liittyy kuitenkin myös joitakin haasteita ja rajoituksia.Ensimmäinen haaste liittyen HTTP/3:n käyttöön on 0-RTT (zero-round-trip-time) yhteydenmuodostuksen riski altistua uusintahyökkäyksille (Replay Attack). Uusintahyökkäyksessä haitallinen toimija voi lähettää uudelleen kaapatun paketin.[11] Toinen HTTP/3:seen liittyvä huolenaihe on, että HTTP/3:n kuljetusprotokolla on rakennettu UDP:n päälle, kaikki palvelut ja sovellukset eivät välttämättä tue HTTP/3:n käyttöä [24]. Kolmas huolenaihe liittyen HTTP/3:n käyttöön, on että HTTP/3 saattaisi olla alttiimpi hajautetuille palvelunestohyökkäyksille (Distributed Denial of Service Attack, DDoS), koska HTTP/3:n tilaton yhteys voi mahdollistaa hyökkäjälle lukuisten pyyntöjen lähettämisen [24].

Kirjallisuus

[1] Trevisan, M. & Giordano, D. & Drago, I. & Khatouni, A. S. Measuring HTTP/3: Adoption and Performance. Teoksessa: *2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)*. 2021, S. 1–8. DOI: 10.1109/MedComNet52149.2021.9501274.

[2] Wendroth, J. & Jaeger, B. A Brief Overview on HTTP. Teoksessa: *Proceedings of the Seminar Innovative Internet Technologies and Mobile Communications (IITM)*. 2022, S. 59–63. DOI: doi: 10.2313/NET-2022-11-1_11.

[3] Nielsen, H. & Fielding, R. T. & Berners-Lee, T. *Hypertext Transfer Protocol – HTTP/1.0*. IETF, 1996. DOI: 10.17487/RFC1945. RFC 1945

[4] Fielding, R. T. & Nielsen, H. & Mogul, J. & Gettys, J. & Berners-Lee, T. *Hypertext Transfer Protocol – HTTP/1.1*. 1997. DOI: 10.17487/RFC2068. RFC 2068

[5] Nielsen, H. & Mogul, J. & Masinter, L. M. & Fielding, R. T. & Gettys, J. & Leach, P. J. & Berners-Lee, T. *Hypertext Transfer Protocol – HTTP/1.1*. 1999. DOI: 10.17487/RFC2616. RFC 2616

[6] Belshe, M. & Peon, R. & Thomson, M. *Hypertext Transfer Protocol Version 2 (HTTP/2)*. 2015. DOI: 10.17487/RFC7540. RFC 7540

[7] Bishop, M. *HTTP/3*. 2022. DOI: 10.17487/RFC9114. RFC 9114

[8] Eddy, W. *Transmission Control Protocol (TCP)*. 2022. DOI: 10.17487/RFC9293. RFC 9293

[9] Postel, J. *User Datagram Protocol*. 1980. DOI: 10.17487/RFC0768. RFC 768

[10] AL-Dhief, F. T. & Sabri, N. & Latiff, N. A. & Malik, N. & Abbas, M. & Albader, A. & Mohammed, M. A. & AL-Haddad, R. N. & Salman, Y. D. & Khanapi, M. Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios. *International Journal of Engineering & Technology.* Vol. 7. 2018. (2018), S. 172–176. Saatavissa: `https://www.researchgate.net/publication/329944111_Performance_comparison_between_TCP_and_udp_protocols_in_different_simulation_scenarios`.

[11] Iyengar, J. & Thomson, M. *QUIC: A UDP-Based Multiplexed and Secure Transport.* 2021. DOI: 10.17487/RFC9000. RFC 9000

[12] Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3.* 2018. DOI: 10.17487/RFC8446. RFC 8446

[13] Bhargavan, K. & Blanchet, B. & Kobeissi, N. Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate. Teoksessa: *2017 IEEE Symposium on Security and Privacy (SP).* 2017, s. 483–502. DOI: 10.1109/SP.2017.26.

[14] Patil, K. *Why Is TLS 1.3 Better And Safer Than TLS 1.2?* Accessed: 30-Aug-2024. 2022. Saatavissa: `https://www.appviewx.com/blogs/why-is-tls-1-3-better-and-safer-than-tls-1-2/`.

[15] Langley, A. & Riddoch, A. & Wilk, A. & Vicente, A. & Krasic, C. & Zhang, D. & Yang, F. & Kouranov, F. & Swett, I. & Iyengar, J. et al. The QUIC Transport Protocol: Design and Internet-Scale Deployment. Teoksessa: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication.* 2017, s. 183–196.

[16] Marx, R. *HTTP/3 From A To Z: Core Concepts.* Accessed: 10-Aug-2024. 2021. Saatavissa: `https://www.smashingmagazine.com/2021/08/http3-core-concepts-part1/`.

[17] Scharf, M. & Kiesel, S. NXG03-5: Head-of-line Blocking in TCP and SCTP: Analysis and Measurements. Teoksessa: *IEEE Globecom 2006.* 2006, s. 1–5. DOI: 10.1109/GLOCOM.2006.333.

[18] Liu, F. & Crowley, P. Security and Performance Characteristics of QUIC and HTTP/3. Teoksessa: *Proceedings of the 10th ACM Conference on Information-Centric Networking.* ACM ICN '23. New York, NY, USA: Association for Computing Machinery, 2023, s. 124–126. DOI: 10.1145/3623565.3623757. Saatavissa: `https://doi-org.libproxy.aalto.fi/10.1145/3623565.3623757`.

[19] Gupta, A. & Bartos, R. Evaluating User Experience of HTTP/3 in Real-World Deployment Scenarios. Teoksessa: *Proceedings of the 2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*. 2022, s. 17–23. DOI: 10.1109/ICIN53892.2022.9758130.

[20] Google. *Performance Scoring*. Chrome Developers Documentation. Viitattu: 21.8.2024. 2019. Saatavissa: `https://developer.chrome.com/docs/lighthouse/performance/performance-scoring`.

[21] Tripathi, P. & Miraz, M. H. & Joshi, S. Comparing Communication Efficiency: HTTP/3 versus HTTP/1.1 Latency in RESTful APIs. Teoksessa: *2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*. 2023, s. 27–31. DOI: 10.1109/CoNTESA61248.2023.10384951.

[22] Perna, G. & Trevisan, M. & Giordano, D. & Drago, I. A First Look at HTTP/3 Adoption and Performance. *Computer Communications* 187 (2022), s. 115–124. DOI: doi.org/10.1016/j.comcom.2022.02.005.

[23] Chellappa, S. & Bartos, R. Is QUIC Quicker with HTTP/3? An Empirical Analysis of Quality of Experience with DASH Video Streaming. Teoksessa: *2022 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. 2022, s. 237–242. DOI: 10.1109/ANTS56424.2022.10227765.

[24] Koch, J. & Falowo, O. & Elrod, N. What We Know About HTTP/3 and Its Implementation: A Literature Review. Teoksessa: *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. 2024, s. 1–7. DOI: 10.1109/ICMI60790.2024.10585883.