

Analiza značajki protokola QUIC transportnog sloja

Pašalić, Luka

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:119:444435>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-30**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences -
Institutional Repository](#)



Zagreb, 23. ožujka 2018.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Tehnologija telekomunikacijskog prometa I**

ZAVRŠNI ZADATAK br. 4575

Pristupnik: **Luka Pašalić (0135238479)**
Studij: **Promet**
Smjer: **Informacijsko-komunikacijski promet**

Zadatak: **Analiza značajki protokola QUIC transportnog sloja**

Opis zadatka:

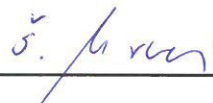
Objasniti karakteristike usluge prijenosa videosadržaja strujanjem. Navesti i ukratko opisati protokole korištene za prijenos videosadržaja strujanjem. Analizirati značajke HTTP adaptivnog strujanja videosadržaja kao i karakteristike novog QUIC protokola transportnog sloja. Opisati utjecaj novog QUIC protokola na isporučenu kvalitetu usluge strujanja videosadržaja.

Mentor:



dr. sc. Marko Matulin

Predsjednik povjerenstva za
završni ispit:



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Luka Pašalić

ANALIZA ZNAČAJKI PROTOKOLA QUIC
TRANSPORTNOG SLOJA

ZAVRŠNI RAD

Zagreb, 2018.

Sveučilište u Zagrebu
Fakultet prometnih znanosti

ZAVRŠNI RAD

**ANALIZA ZNAČAJKI PROTOKOLA QUIC
TRANSPORTNOG SLOJA**

**CHARACTERISTICS OF THE TRANSPORT LAYER
PROTOCOL QUIC**

Mentor: doc.dr. sc. Marko Matulin

Student: Luka Pašalić

JMBAG: 0135238479

Zagreb, rujan 2018.

ANALIZA ZNAČAJKI PROTOKOLA QUIC TRANSPORTNOG SLOJA

SAŽETAK

Kako računalna tehnologija svakodnevno napreduje već dugi niz godina, tako su se i povećali zahtjevi korisnika za uslugama. U smislu videosadržaja, više nije dovoljno samo isporučiti videosadržaj, kako bi korisnici bili zadovoljni. Korisnici su povećali svoje zahtjeve, kako u smislu kvalitete tako i u upravljanju videosadržaja. Zbog svega navedenoga, razvili su se protokoli koji su omogućili osnovne operacije pri reprodukciji videosadržaja te protokoli koji se prilagođavaju raznim zaslonima i stanju prijenosnog kapaciteta mreže. Iz razloga što postoji konkurencija, mnogo proizvođača želi implementirati svoje protokole u razne sustave, što je krajnjim korisnicima omogućilo bolju kvalitetu usluge, jer svi nastoje implementirati protokole koji bi prenosili videosadržaj strujanjem sa što manje poteškoća. Jedan od posljednjih razvijenih protokola koji je još u fazi izrade, ali se implementirao u razne sustave je QUIC (engl. *Quick UDP Internet Connectins*) protokol, s kojim se pokušava dodatno smanjiti kašnjenje pri prijenosu videosadržaja strujanjem.

KLJUČNE RIJEČI: videosadržaj; protokol; prijenos videosadržaja strujanjem, QUIC

SUMMARY

As computer technology improves, from one day to another, the users have increased their demands about services. In case of a video, it is not enough to just deliver a video to the users. Users increased their demands, they want better quality video with built-in control operations. Because of these reasons, the protocols are started to develop, with basic functions when reproducing video and protocols which can adapt to various displays and available bandwidth of a user. Because of a competition, a lot of manufacturers wants to implement their protocol in various systems, which gives users better quality of service, because everyone wants to implement a protocol that will stream a video with as low rate of difficulties as possible. One of the latest developed protocols, which is still in a state of a development, is QUIC (engl. *Quick UDP Internet Connectins*) protocol, which tries to additionally reduce the latency when streaming video.

KEY WORDS: video; protocol; video streaming; QUIC

SADRŽAJ

1. UVOD	1
2. ZNAČAJKE USLUGE PRIJENOSA VIDEOSADRŽAJA STRUJANJEM.....	3
3. KORIŠTENI PROTOKOLI ZA PRIJENOSA VIDEOSADRŽAJA.....	5
3.1. RTSP	6
3.2. RTP	8
3.3. RTMP.....	9
4. HTTP ADAPTIVNO STRUJANJE VIDEOSADRŽAJA	11
4.1. ADAPTIVNO STRUJANJE.....	11
4.2. TEHNOLOGIJE HTTP ADAPTIVNOG STRUJANJA	14
4.2.1. Apple HTTP Live Streaming (HLS).....	14
4.2.2. Microsoft Silverlight Smooth Streaming (MSS).....	15
4.2.3. Adobe HTTP Dynamic Streaming (HDS).....	16
4.2.4. Dynamic Adaptive HTTP Streaming (DASH).....	17
5. KARAKTERISTIKE NOVOG QUIC PROTOKOLA	19
5.1. KLJUČNE PREDNOSTI.....	19
5.1.1. Latencija pri uspostavljanju veze	19
5.1.2. Fleksibilna kontrola zagušenja.....	20
5.1.3. Strujanje i kontrola toka	21
5.1.4. Multipleksiranje.....	21
5.1.5. Autentifikacija i enkripcija	22
5.1.6. Forward Error Correction (FEC)	22
5.1.7. Migracija veze.....	22
5.2. PAKETIZACIJA I POUZDANOST	23
5.2.1. Procesiranje paketa i ACK.....	23
5.2.2. Ponovno slanje informacija.....	24
5.2.3. Veličina paketa	24
5.2.4. Path Maximum Transmission Unit (PMTU)	24
5.2.5. Tipovi paketa i format zaglavlja	25
5.3. QUIC STRUJANJE	26
5.4. ZAVRŠETAK VEZE	27
6. UTJECAJ NA KVALITETU USLUGE	29
7. ZAKLJUČAK.....	32
LITERATURA	33

POPIS KRATICA	35
POPIS SLIKA	37
POPIS TABLICA.....	38

1. UVOD

QUIC (engl. *Quick UDP Internet Connections*) je multipleksirani i sigurni transportni protokol koji radi iznad UDP-a (engl. *User Datagram Protocol*). Njegov glavni cilj je pružiti fleksibilan komplet usluga, koje mu omogućuju da bude siguran protokol za mnoge aplikacije te smanji kašnjenje u usporedbi s drugim protokolima. Kako je zadnjih godina glavna zadaća Interneta informiranje u najbržem mogućem vremenu te pregled videosadržaja, vrlo je važno zadovoljiti korisnikove zahtjeve. Jedan od glavnih zahtjeva, koji korisniku daje negativan ili pozitivan stav o usluzi je vrijeme učitavanja (videozapisa, stranice itd.).

Posljednjih godina, pokušava se na različite načine ubrzati pretraživanje i korištenje interneta. Pružatelji usluga pokušavaju smanjiti vrijeme učitavanja tako što povećavaju prijenosni kapacitet te korištenjem brzih mrežnih uređaja. Oni koji razvijaju web pretraživače pokušavaju napraviti *software* efikasnijim. Google-ov nedavni pristup takvom problemu je taj, da se pokuša smanjiti kašnjenje u transportnom sloju OSI (engl. *Open System Interconnection*) referentnog modela. Kako su dominantni protokoli UDP i TCP (engl. *Transmission Control Protocol*), od kojega svaki ima svoje prednosti i nedostatke, Google razvija novi protokol, s kojim pokušava objediniti prednosti jednoga i drugoga, koji se naziva QUIC.

Cilj i svrha ovog završnog rada je prikazati prednosti i mane navedenoga protokola, kao i pojasniti ostale protokole koji izvršavaju iste ili slične zadatke te koji su to zadatci koji se izvršavaju.

Rad se sastoji od sedam poglavlja:

1. Uvod
2. Značajke usluge prijenosa videosadržaja strujanjem
3. Korišteni protokoli za prijenos videosadržaja
4. HTTP adaptivno strujanje videosadržaja
5. Karakteristike novog QUIC protokola
6. Utjecaj na kvalitetu usluge
7. Zaključak.

U drugom poglavlju opisuju se značajke prijenosa videosadržaja strujanjem te važnost takve usluge na cjelokupni mrežni promet.

Treće poglavlje navodi i opisuje osnovne značajke protokola koji se u najčešćim slučajevima koriste za prijenos videosadržaja strujanjem.

Četvrtim poglavljem pojašnjava se posebna tehnologija adaptivnog strujanja videosadržaja, zašto je tako bitna kod kvalitete usluge, vrste takvih tehnologija te prednosti i mane svake vrste.

Poglavlje broj pet bavi se značajkama novog QUIC protokola, koje su njegove glavne prednosti te način na koji pospješuje određene funkcije protokola, koji se bave takvim procesima.

Šesto poglavlje donosi istraživanje, kojim se utvrđuje u kakvim okolnostima QUIC protokol ostvaruje bolje rezultate (u smislu parametara kvalitete usluge) od ostalih sličnih protokola.

Sedmo poglavlje donosi krajnje zaključke na temelju svega navedenoga u ovome radu.

2. ZNAČAJKE USLUGE PRIJENOSA VIDEOSADRŽAJA STRUJANJEM

Popularnost prijenosa videosadržaja strujanjem se konstantno povećava, kako ovoga tako i prethodnog desetljeća. Nedavne studije navode kako je udio internetskog prometa, koji zauzimaju videozapisi strujanjem, 2015. godine iznosio 63% sveukupnog prometa internetom, [17]. Cisco VNI (engl. *Visual Networking Index*) Forecast predviđa povećanje na 79% do 2020. godine, [17]. Dva su dominantna izvora za prijenos videozapisa strujanjem u Sjevernoj Americi, Netflix i Youtube. Youtube je najpopularniji izvor strujanja videozapisa u Europi i Južnoj Americi, [1].

Iako je navedena popularnost iznimno visoka, malo se zna o strategijama koje koriste dva navedena i najveća izvora kako bi prenosili videozapise strujanjem. Strujanje videozapisa putem internetske mreže omogućuje korisnicima pokretanje videozapisa dok je videozapis u fazi preuzimanja. Korisnici mogu gledati videozapise koristeći web preglednike, na mobilnim uređajima ili na mobilnim aplikacijama. Mobilna aplikacija je nativna Netflix ili Youtube aplikacija koja se pokreće na mobilnim uređajima, [1]. U ovome poglavlju u obzir će se uzeti dvije navedene aplikacije pokretane na iOS i Android operativnim sustavima.

Youtube, jedna od najpopularnijih stranica za pokretanje i gledanje videozapisa objavljenih od strane drugih korisnika, podržava dva kontejnera za strujanje videa, Adobe Flash i HTML5 (engl. *HyperText Markup Language*). Adobe Flash je zadani kontejner kada se na Youtube pristupa preko osobnog računala. Korisnici moraju instalirati dodatak kako bi gledali Adobe Flash videozapise. HTML5 podržava videozapise koji ne zahtijevaju nikakve dodatke. HTML5 je zadani kontejner kada se na Youtube pristupa preko mobilnih aplikacija putem iOS-a i Android-a. Zadani kontejner za videozapise visoke definicije je Adobe Flash. Netflix koristi Microsoft-ov Silverlight za strujanje videozapisa. Tijekom uobičajene sesije strujanja, osim sadržaja videozapisa, poslužitelji strujanja šalju dodatne podatke. Na primjer, dodatni podatci sadržavaju detalje o povezanim videozapisima te reklamne sadržaje, [1].

S točke gledišta teleprometnog inženjerstva razlikuju se tri vrste videoaplikacija te je prijenos videosadržaja strujanjem samo jedna od njih, postoje još

video na zahtjev te video telefonija i videokonferencija. Videoaplikacije se razlikuju u bitnim značajkama kao što su:

- asimetrije u prijenosu,
- zahtijevane veličine kapaciteta linka,
- osjetljivost na kašnjenje i kolebanje kašnjenja.

Trenutni i videoizvori koji tek trebaju doći, bitno utječu na dizajniranje i planiranje mreže, koja originalno nije bila zamišljena kako bi se preko nje prenosili videosadržaji. Prijenos videozapisa strujanjem sadrži sljedeće značajke te ih je potrebno razlikovati:

- radi li se o prijenosu videozapisa strujanjem u realnom vremenu ili ne (engl. *real time or non-real time*);
- zahtijeva li se konstantna brzina prijenosa CBR (engl. *Constant Bit Rate*) ili varijabilna brzina prijenosa VBR (engl. *Variabile Bit Rate*);
- radi li se o prijenosu prema jednoj točki (engl. *Unicast*) ili prijenosu prema određenom većem broju točaka (engl. *Multicast*).

U slučaju da se radi o stvarnovremenskom prijenosu, videozapis se snima, zatim komprimira i odmah šalje na prijenos, za razliku od prijenosa videozapisa kada se ne zahtijeva prijenos u stvarnom vremenu, već se snimljeni videosadržaj komprimira i sprema na poslužitelj videa. Kada korisnik u nekom trenutku zahtijeva takav sadržaj, on se prije slanja konvertira u IP (engl. *Internet Protocol*) pakete. Format prijenosa za aplikacije prijenosa videosadržaja strujanjem u stvarnom vremenu i onoga videosadržaja koji nisu stvarnovremeni, ne mora biti različit ako je brzina prijenosa onih koji sadržaj gledaju ista, [2].

Kada se spominje varijabilna i konstanta brzina prijenosa, u kontekstu prijenosa videosadržaja strujanjem, može se reći da jednostavni prijenos videa i nekompromirani videozapisi mogu biti zadovoljeni konstantnom brzinom prijenosa. Većina shema komprimiranja videozapisa proizvodi promet koji zahtijeva varijabilnu brzinu prijenosa, iz razloga što se komprimiranjem videozapisa postiže prenošenje samo onih uzoraka koji se mijenjaju. Veličina zahtijevanog kapaciteta u pojedinom trenutku ovisiti će o sadržaju kojega korisnik gleda, jesu li promjene na slikama videozapisa velike ili male, [2].

3. KORIŠTENI PROTOKOLI ZA PRIJENOS VIDEOSADRŽAJA

Prije nego što se navedu protokoli korišteni za prijenos videosadržaja strujanjem, treba definirati što to protokoli za prijenos videosadržaja strujanjem jesu. Većina digitalnih videozapisa su dizajnirani za dvije stvari, pohranu i reprodukciju. Većina videozapisa nisu dizajnirani za strujanje, jednostavnije rečeno, strujanje videozapisa je proces koji uključuje razdvajanje videozapisa u manje dijelove (engl. *Chunks*). Ti manji dijelovi se šalju sekvencijalno i pokreću se kada se zaprime. U slučaju strujanja videozapisa uživo, izvor videozapisa dolazi izravno iz kamere, u suprotnom, dolazi iz datoteke, [3].

Dok je kroz navedeno opisana osnovna verzija procesa, protokoli za strujanje videozapisa mogu biti poprilično kompleksni. Postoje i protokoli koji se prilagođavaju brzini prijenosa (engl. *Adaptive bitrate protocols - ABR*). Ova tehnologija omogućuje gledatelju videosadržaja najbolju moguću kvalitetu koju njegov sustav u tom trenutku može izdržati. Neki protokoli se usredotočuju na smanjenje kašnjenja, kako bi trenutni događaj što prije bio prikazan na gledateljevom zaslonu, [3].

Biranje protokola za strujanje videozapisa može biti težak zadatak koji ovisi o tipu podataka koji se trebaju prenijeti. Komunikacija mora biti uspostavljena korištenjem protokola, koji je formiran od skupine pravila, koja definiraju kako se podatci prenose putem mreže te kako se dijele na manje dijelove kao što su zaglavlje, podatci, autentifikacija i rukovanje greškama. Zbog svega navedenoga, protokoli za strujanje videozapisa mogu se promatrati kao komunikacijski protokoli gdje su podatci koji se prenose, medijski podatci, [4].

Koncentracija ovog poglavlja usmjerena je istraživanju protokola koji će najefikasnije, prenijeti videosadržaj. Zbog toga razloga, najvažnija točka je garancija maloga kašnjenja te efikasnog prijenosa sa ponekim gubitkom paketa. Tradicionalno, ovi protokoli se dijele na *push-based* i *pull-based* protokole, [4].

Push-based protokoli sastoje se od uspostavljene komunikacije između klijenta i servera, gdje je klijent odgovoran za uspostavljanje konekcije, a poslužitelj šalje pakete sve dok klijent ne stopira ili ne omete komunikaciju. U ovome tipu protokola, poslužitelj, osim što šalje medijske pakete, održava sesiju kako bi slušao

naredbe klijenta. Ovi protokoli u pravilu rade preko UDP-a, ali mogu raditi i preko TCP-a ukoliko je potrebno, [4].

Pull-based protokoli su bazirani na HTTP (engl. *Hyper Text Transfer Protocol*) protokolu i sastoje se od komunikacije između klijenta i servera, gdje je klijent odgovoran za slanje zahtijeva poslužitelju, a poslužitelj započinje komunikaciju gdje klijent preuzima strujanje videozapisa. U ovim protokolima, brzina strujanja ovisi i o širini prijenosnog pojasa (engl. *bandwidth*) mreže klijenta. Najčešće korišten *pull-based* protokoli se povezuju pod pojmom *progressive download*, gdje klijent šalje zahtjev poslužitelju i počinje preuzimati medijski sadržaj što je brže moguće, [3].

Tablica 1. Usporedba *push* i *pull-based* protokola za strujanje videosadržaja

	Izvor	Protokoli	Iskoristivost prijenosnog pojasa	<i>Multicast</i> podrška
<i>Push-based</i>	Poslužitelji i televizijske kuće	RTSP, RTP, UDP	Više efikasna	Da
<i>Pull-based</i>	Web poslužitelji	HTTP	Manje efikasna	Ne

Izvor: [4]

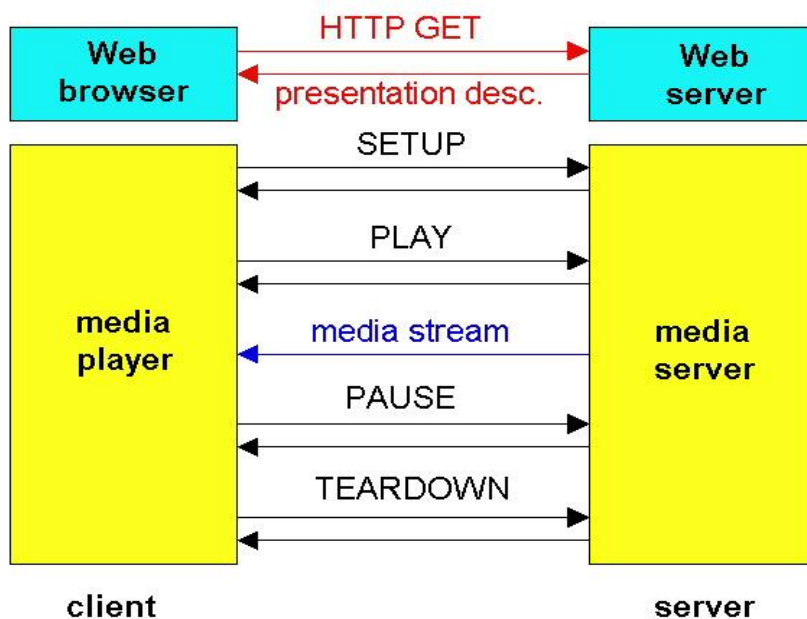
3.1. RTSP

Real Time Streaming Protocol (RTSP) je nekonekcijski protokol koji ispunjava funkcije aplikacijskog sloja OSI referentnog modela, upravlja i kontrolira prijenos stvarnovremenskih podataka, kao što su audio i videozapisi. RTSP je jedan od najtradicionalnijih protokola koji je izrađen sredinom devedesetih godina 20. stoljeća. Izvori podataka uključuju oboje, podatke koji se stvaraju uživo te podatke koji su pohranjeni na raznim medijima za pohranu. Protokol je korišten kako bi se uspostavila i kontrolirala medijska sesija od-kraja-do-kraja (engl. *end-to-end*), pritom služeći kao mrežni upravljač za vremenski sinkronizirane, kontinuirane videosadržaje koji se prenose, [4].

RTSP je primjer *pull* protokola iz razloga što je moguće kontrolirati naredbe kao što su pokreni, zaustavi ili pauziraj sadržaj (slika 1). Dizajniran je da kontrolira višestruke sesije od-kraja-do-kraja, i dozvoljava odabir kanala kojim će se sadržaj dostaviti, [4].

Sintaksa RTSP-a podržava sljedeće operacije:

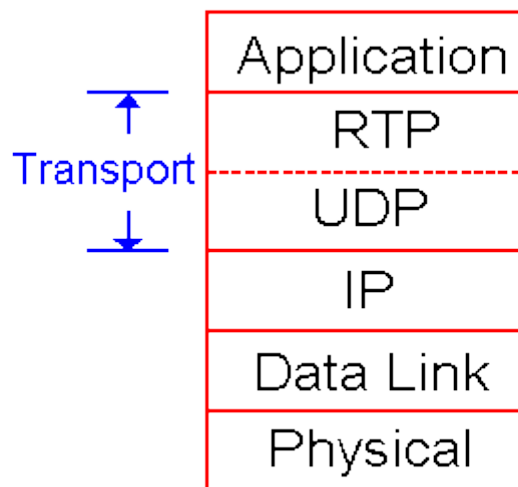
- povlačenje sadržaja s poslužitelja; korisnik može zatražiti opis prezentacije preko HTTP-a ili neke druge metode. Ako je prezentacija namijenjena prosljeđivanju na više točaka (engl. *Multicast*), opis prezentacije će sadržavati adrese i portove točaka kojima se ona prosljeđuje. Ako je prezentacija predodređena za slanje na samo jednu točku (engl. *Unicast*), korisnik ustupa adresu odredišta iz sigurnosnih razloga,
- poziv medijskog poslužitelja (engl. *Server*) u konferenciju; medijski poslužitelj može biti pozvan da se pridruži postojećoj konferenciji, bilo da reproducira sadržaj ili snima sadržaj. Ovakav način koristan je za distribuciju aplikacija za učenje. Više korisnika koji sudjeluju u konferenciji može se izmjenjivati koristeći kontrolne tipke,
- dopuna medija na postojećoj prezentaciji; naročito za prezentacije uživo, korisno je poslužiteljima da imaju mogućnost informiranja korisnika o raspoloživosti dodatnog medijskog sadržaja, [4].



Slika 1. Interakcija između klijenta i poslužitelja korištenjem RTSP-a, [6]

3.2. RTP

RTP uobičajeno radi iznad UDP-a. Dijelovi podataka u kojemu se nalazi videosadržaj, koji je generiran od strane pošiljatelja, enkapsuliraju se u RTP pakete i svaki paket je enkapsuliran u UDP segment. Iz razloga što RTP pruža usluge (vremenske oznake, brojeve redoslijeda itd.) multimedijским aplikacijama, može se promatrati kao podsloj transportnog sloja (slika 2.), [6].



Slika 2. RTP kao podsloj transportnoga sloja, [6]

Iz perspektive programera za razvoj aplikacija, RTP nije dio transportnog sloja već aplikacijskog sloja. Ovakva podijeljenost se događa iz razloga što programer mora integrirati RTP u aplikaciju. Za pošiljateljsku stranu aplikacije, programer mora napisati kôd koji stvara RTP enkapsulirajuće pakete, tada ih aplikacija šalje u UDP sučelje. Slično, na prijemnoj strani aplikacije, RTP paketi ulaze u aplikaciju preko UDP sučelja. Programer tada mora napisati kôd u aplikaciju kako bi se izvukli dijelovi medijskog zapisa iz RTP paketa, [6].

Treba razjasniti da RTP ne pruža mehanizme koji osiguravaju dostavljanje podataka u određenom vremenu ili da osigurava garanciju kvalitete usluge. RTP čak ni ne osigurava sigurnu dostavu paketa ili očuvanje njihovog redoslijeda. Ono što RTP omogućava, je da svakom izvoru (na primjer, kameri) bude dodijeljeno RTP neovisno strujanje paketa. Na primjer, u slučaju videokonferencije između dva sudionika, četiri RTP strujanja mogu biti aktivna; dva za prijenosa videozapisa u oba smjera te dva za prijenos audio zapisa u oba smjera, [6].

RTP paketi nisu ograničeni na aplikacije koje podržavaju prijenos prema samo jednoj točki. Paketi kod ovoga protokola, mogu biti poslani iz jedne točke prema više točaka ili od više točaka prema više točaka . Za sesije gdje se koristi prijenos od više točaka prema više točaka, svi pošiljatelji i izvori u sesiji uobičajeno šalju svoja RTP strujanja, prema istome *multicast* stablu (engl. *multicast tree*) s istom *multicast* adresom, [6].

3.3. RTMP

RTMP je protokol koji ispunjava funkcije aplikacijskog sloja OSI referentnog modela te koji dostavlja medijske zapise, kako na zahtjev tako i uživo, Adobe Flash aplikacijama. RTMP podržava videozapise u MP4¹ (MPEG-4) i FLV (engl. *Flash Video*) formatima i audio zapise u MP3 (engl. *MPEG-1 audio layer 3*) i AAC (engl. *Advanced Audio Coding*) formatima. Pruža nekoliko prednosti pri usporedbi s HTTP-om, kao što su robusna podrška za strujanje uživo i dinamično strujanje, gdje se kvaliteta videozapisa prilagođava trenutno raspoloživom prijenosnom kapacitetu mreže. Aplikacije koje reproduciraju videozapise s ovakvim protokolom, mogu ubrzati videozapis na kraj, što je izrazito korisno kod videozapisa velike duljine, [5].

Prema specifikacijama [7], ovaj protokol ima više varijacija:

- „obični“ RTMP,
- RTMPS - RTMP preko TLS/SSL (engl. *Transport Layer Security/Secure Socket Layer*) konenkcije,
- RTMPE - RTMP enkriptiran putem Adobe-ovih sigurnosnih mehanizama,
- RTMPT – enkapsuliran zahtjevima HTTP-a da prolazi kroz vatrozid.

Karakteristika RTMP je da izbjegne kašnjenje u komunikaciji, dostavljajući videozapise bez većih poteškoća i razdvajajući ih u fragmente, te ih umetati i multipleksirati² putem jedne konekcije. Multipleksiranje je obavljeno na razini paketa, s RTMP paketima preko nekoliko različitih aktivnih kanala te se šalju na takav način da se osigura da svaki kanal bude zadovoljen u smislu njegovog prijenosnog kapaciteta, kašnjenja i ostalih parametara kvalitete usluge (engl. *QoS – Quality of*

¹ .mp4 – medijski format, pohranjuje videozapis, audiozapis, podnapise i slike

² Multipleksiranje – proces dijeljenja fizičkog medija u svrhu prenošenja višestrukih tokova informacija sa više izvora

Service). RTMP definira nekoliko virtualnih kanala na koje bi se paket mogao poslati ili biti primljen i koji djeluju neovisno jedan o drugome. Tijekom uobičajene RTMP sesije, nekoliko kanala mogu biti aktivni istovremeno u bilo kojem vremenu, [7].

RTMP ima nekoliko direktnih i indirektnih nedostataka:

- namijenjen je da se koristi zajedno s *Flash playerom*,
- kako RTMP šalje podatke pokretaču, tako prijenosni kapacitet mora biti veći nego što je količina podataka koju videozapis generira, u suprotnome strujanje videozapis će zastajati,
- mnoge medijske tvrtke su u fazi tranzicije s Flash/RTMP na HTTP *adaptive bitrate* protokole, zbog sigurnosnih nedostataka tijekom godina,
- *flash* platforma nikada nije podržana od strane tvrtke Apple za svoje iOS proizvode, [5].

4. HTTP ADAPTIVNO STRUJANJE VIDEOSADRŽAJA

Adaptivno strujanje je relativno nedavna tehnika, koja služi kako bi se strujao videosadržaj preko IP mreže. Starije tehnologije strujanja videosadržaja su se bazirale na protokolima kao što su RTP, RTSP i RTMP. Današnje adaptivno strujanje videosadržaja se bazira na HTTP protokolu, osiguravajući efikasnu dostavu medijskog sadržaja preko globalne distribucijske HTTP mreže, kao što je Internet. HTTP je osnovni podatkovno komunikacijski protokol korišten za *World Wide Web* (drugi standard koji kontrolira kako *World Wide Web* radi je HTML, koji se brine kako su web stranice formatirane i prikazane), [5].

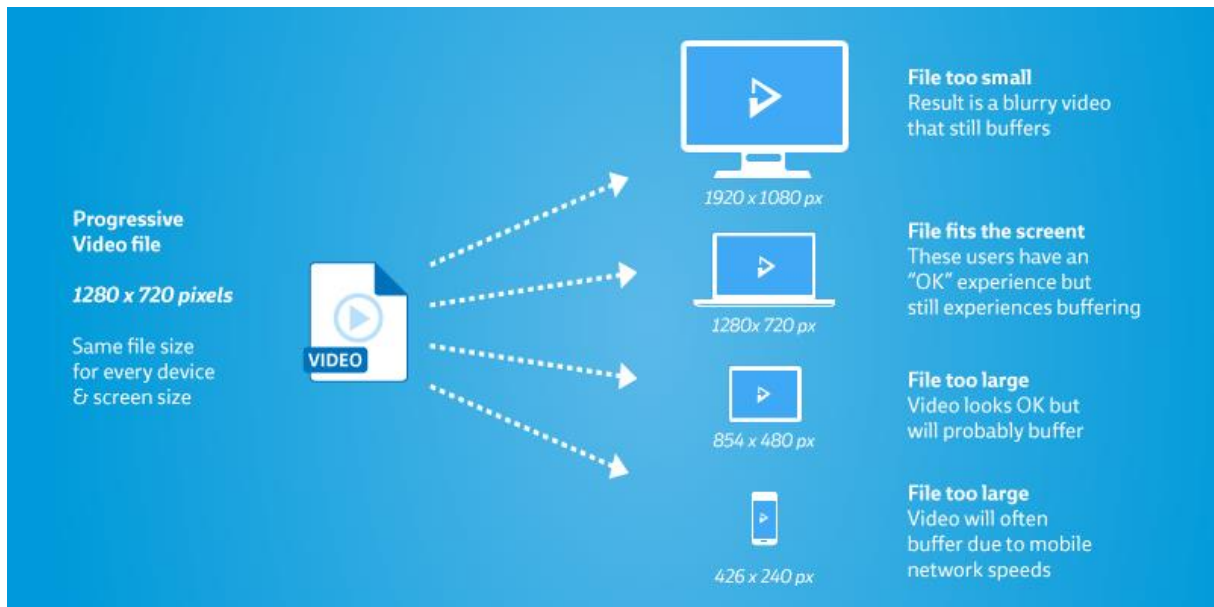
HTTP obavlja funkcije aplikacijskog sloja OSI referentnog modela i upotrebljava TCP protokol, koji obavlja funkcije transportnog sloja, gdje se TCP brine o pouzdanoj dostavi paketa. HTTP također koristi stvarnovremenske protokole kao što je UDP, kojemu je prioritet nisko kašnjenje, a ne pouzdana dostava svakoga paketa kao što je slučaj kod TCP-a. HTTP radi kao protokol kojemu se šalje zahtjev za odgovor u modelu klijent - poslužitelj. Dizajniranjem HTTP-a dozvoljeno je mrežnim elementima koji se nalaze između ova dva sudionika da poboljšaju ili omoguće komunikaciju između klijenta i poslužitelja, [5].

4.1. ADAPTIVNO STRUJANJE

Adaptivno strujanje je tehnologija dizajnirana kako bi se dostavljao videozapis korisnicima na najefikasniji način. Kako bi se objasnilo što adaptivno strujanje je, objasniti će se što adaptivno strujanje nije, odnosno što je progresivno strujanje videozapisa. Progresivno strujanje videozapisa je strujanje gdje se jedan videozapis šalje preko interneta. Tip te datoteke je često .mp4 , ali može biti i neki drugi format. Videozapis koji se šalje korištenjem tehnologije progresivnog strujanja, može biti raširen ili se može suziti kako bi odgovarao različitim zaslonima te nebitno na kojemu se uređaju pokreće, datoteka tog videozapisa će uvijek biti ista, [8].

Glavni fokus slike 3, je taj da je se videozapis ne mijenja bez obzira na uređaj koji se koristi za reprodukciju. Za primjer se uzeo videozapis rezolucije 1280 piksela x

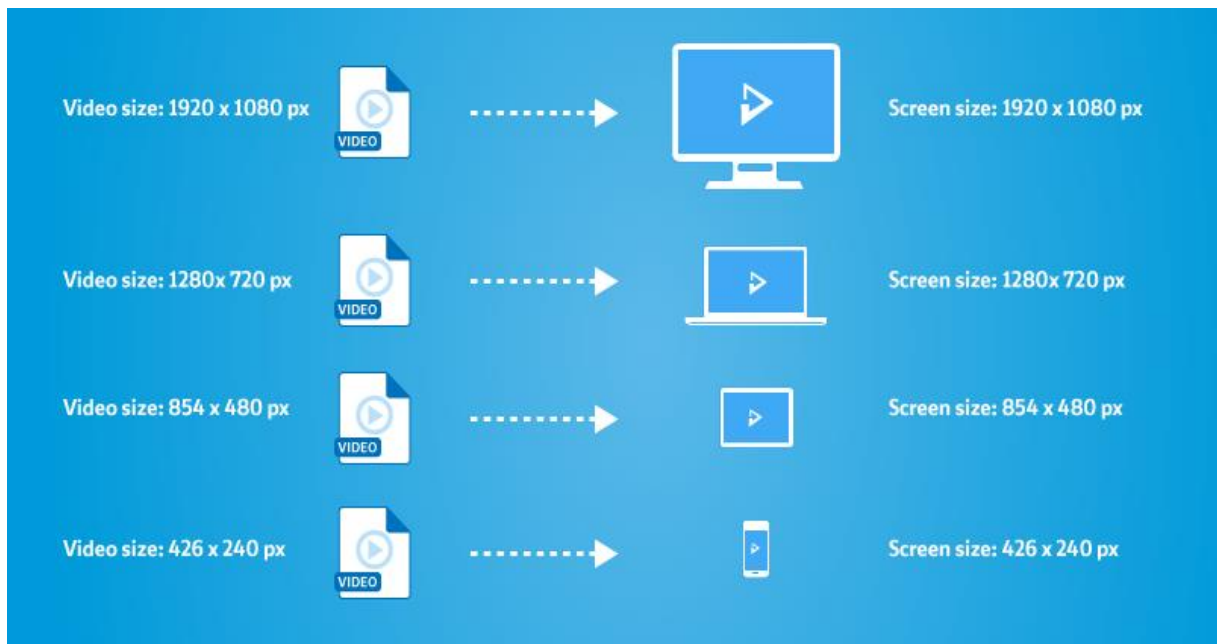
720 piksela, te će taj videozapis biti sužen ili raširen kako bi se prilagodio zaslonu, [8].



Slika 3. Put videozapisa od poslužitelja do korisnika progresivnim strujanjem, [8]

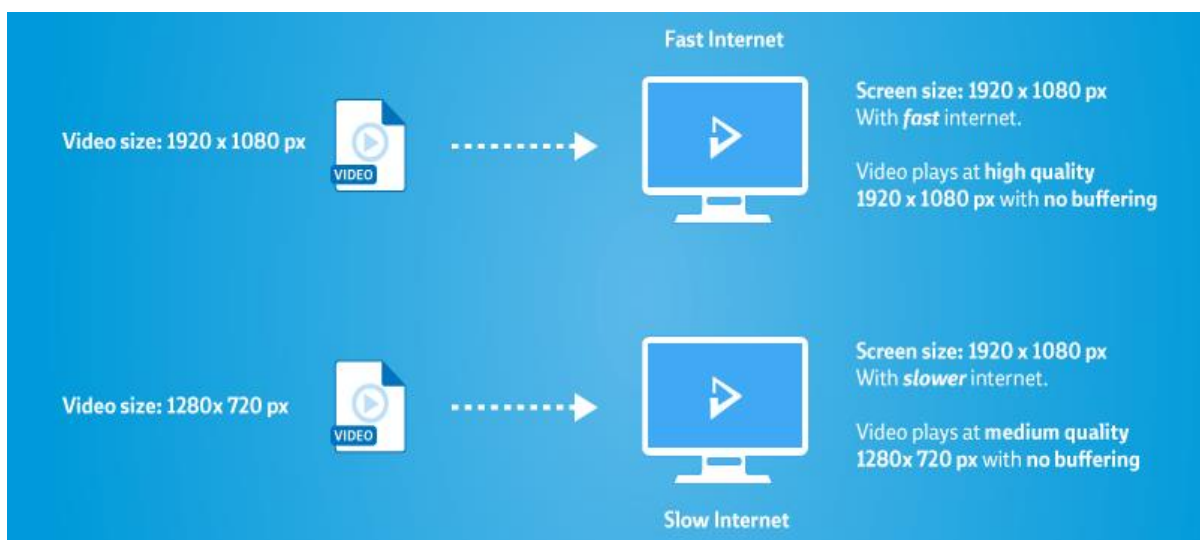
Postoje dva problema koja se događaju pri ovome tipu strujanja. Prvi je kvaliteta, videozapis koji je rezolucije 1280 piksela x 720 piksela se nikada neće reproducirati odgovarajućom kvalitetom na zaslonu veće rezolucije. Videozapis će se raširiti te će biti vidljivi pikseli. Drugi problem je zastajkivanje videozapisa (engl. *Buffering*). Ako korisnik ima internetsku vezu lošije kvalitete i ne može preuzeti videozapis odgovarajućom brzinom, videozapis će imati potrebu za zaustavljanjem, pričekati još podataka i tek tada se pokrenuti. Navedena situacija se može događati vrlo često, pogotovo u mobilnoj mreži, gdje kvaliteta veze ovisi o lokaciji korisnika i njegovog uređaja te ostalim parametrima, [8].

Adaptivno strujanje efikasno rješava oba navedena problema i to vrlo jednostavno. Adaptivno strujanje dopušta opskrbljivaču videozapisa da izradi različite videozapise za zaslone različitih rezolucija. Slika 4 jasno pokazuje kako adaptivno strujanje radi. Kao što se može vidjeti, može se strujati različiti videozapis za različite zaslone te tako biti siguran da će kvaliteta zaprimljenog videozapisa biti u redu, [8].



Slika 4. Adaptivno strujanje, različiti videozapisi za različite zaslone, [8]

Drugi problem je zastajkivanje videozapisa. Većina videozapisa se reproducira koristeći dvadeset i četiri sličice u sekundi, pa tako korisnikova internetska veza mora preuzeti dvadeset i četiri sličice u sekundi kako bi se izbjeglo zastajkivanje. Adaptivno strujanje može riješiti taj problem tako što se prilagođava korisnikovoj vezi. Manji videozapis može se preuzeti brže od velikoga, pa ako korisnik ima sporu vezu sa internetom, adaptivno strujanje će preuzeti manji videozapis kako bi se reprodukcija videozapisa nastavila bez zastajkivanja.



Slika 5. Adaptivno strujanje, prilagođavanje na različite brzine veze, [8]

Iz slike 5 može se vidjeti da je prioritet za korisnika da izbjegne zastajkivanje, puno prije nego održavanje kvalitete videozapisa u smislu rezolucije. Korisnik će biti zadovoljniji ako gleda sadržaj nekoliko minuta u lošijoj kvaliteti nego da zbog spore internetske veze, gleda u ikonu učitavanja te čeka preuzimanje videozapisa bolje kvalitete, [8].

Kako se veza može mijenjati iz sekunde u sekundu, kvaliteta videozapisa se može mijenjati s jedne na drugu. Navedena operacija je izvediva korištenjem segmenata. Kada se videozapis kodira za prilagođeni format, razdvaja se u segmente. To su kratki videozapisi, obično duljine od četiri sekunde, ali mogu biti i duži i kraći. Na kraju svakoga segmenta, program koji reproducira videozapis može promijeniti datoteku videozapisa s koje povlači sadržaj. Adaptivno strujanje nije samo operacija koja je odlična za mobilne mreže kada se mijenja lokacija. Internet je sam po sebi vrlo kompleksna mreža i performanse tih sustava se konstantno mijenjaju, [8].

4.2. TEHNOLOGIJE HTTP ADAPTIVNOG STRUJANJA

Kako bi se osigurala optimalna kvaliteta unatoč uvijek zahtjevnoj prirodi strujanja videozapisa te nedostatku kvalitete usluge u neuređenim mrežama, Apple, Microsoft, Adobe i MPEG su razvili adaptivne protokole za dostavu videozapisa. Oni su prihvaćeni od strane operatora i pružatelja usluga. Ovo potpoglavlje će razmotriti četiri glavne tehnologije za HTTP adaptivno strujanje:

- Apple-ov HTTP Live Streaming (HLS),
- Microsoft-ov Silverlight Smooth Streaming (MSS),
- Adobe-ov HTTP Dynamic Streaming,
- MPEG-ov Dynamic Adaptive Streaming over HTTP (DASH).

4.2.1. Apple HTTP Live Streaming (HLS)

HLS koristi MPEG-2 pri strujanju te segmentira videozapise u niz MPEG-2 TS³ datoteka koje enkapsuliraju audio i videozapise. Ti segmenti su pohranjeni na bilo kojemu HTTP poslužitelju zajedno s datotekom s popisima (engl. *Playlist file*).

³ TS datoteke – format videozapisa koji komprimira videozapise korištenjem MPEG kompresije

Datoteka s popisima je tekstualna datoteka s m3u8⁴ ekstenzijom. HLS definira dva tipa datoteka s popisima, normalne i varijabilne. Normalne prikazuju URL-ove koji ukazuju na dijelove videozapisa koji trebaju biti reproducirani sekvencijalno. Varijabilne ukazuju na kolekciju različitih normalnih datoteka s popisima, jedna za svaki izlazni profil, [9].

HLS je podržan od strane web pretraživača namijenjenih za osobna računala, pa tako i Android i iOS uređaja. HTML5 pokretači videozapisa također podržavaju HLS. To omogućava videozapisu koji se prenosi strujanjem, dostizanje koliko god je gledatelja moguće. HLS je najrašireniji protokol današnjice. Istraživanja pokazuju da se samo 8,2 % gledatelja vraća na stranicu na kojoj su imali problematično iskustvo s gledanjem videozapisa, [3].

Prednosti HLS-a:

- jednostavan protokol te ga je lako mijenjati,
- korištenjem TS datoteka ukazuje da postoji bogati sustav za testiranje i pokazivanje sukladnosti datoteka,
- zarade putem HLS-a se lakše ostvaruju jer je uobičajeni protokol za iOS uređaje koji su naviknuti na plaćanje aplikacija i ostalih usluga.

Nedostatci HLS-a:

- HLS nije podržan od strane Windows-a,
- HLS je sustav u kojemu različiti iOS korisnici imaju različite sposobnosti te to limitira vrijednost budućeg napretka u HLS-u. [9].

4.2.2. Microsoft Silverlight Smooth Streaming (MSS)

2008. godine Microsoft je objavio da će izdati novi IIS 7.0 (engl. *Internet Information Server*) sa novim adaptivnim HTTP strujanjem te će se zvati *Smooth Streaming*, kojemu će glavni prioritet biti dostava videozapisa visoke definicije bez poteškoća. Klijent baziran na Silverlight-u, neprestano identificira raspoloživi prijenosni kapacitet, iskorištenje procesorske jedinice te rezoluciju zaslona na kojemu se videozapis pokreće, kako bi odredio najbolji odgovarajući tip sadržaja koji će se strujati, [10].

⁴ m3u8 – m3u8 datoteke su baza za HLS-ov format videozapisa

Microsoft je demonstrirao sposobnosti svojega sustava za strujanje videosadržaja na nekoliko velikih događanja kao što su ljetne i zimske olimpijske igre 2008. i 2010. godine te teniski turnir u Wimbledon-u 2009. godine. Prenoseći zimske olimpijske igre 2010. godine u Vancouver-u, NBC (američka radijska i televizijska mreža) je koristio HTTP strujanje (*smooth streaming*) kako bi pružio online strujanje za preko 50 milijuna korisnika u 720p rezoluciji te tako stvorivši promet od 3,6 petabajta, [10].

Microsoft uključuje iskorištenje procesorske jedinice kao indikator za strujanje videozapisa, što je posebno korisno pri gledanju takvoga sadržaja na pametnome telefonu ili tablet uređaju. To znači da indikator iskorištenja procesora reagira na način da ako je iskorištenje veliko, smanjuje se kvaliteta videozapisa koji se prenosi strujanjem te tako smanjuje iskorištenje procesora, [10].

Prednosti MSS-a:

- IIS stvara formate za strujanje u kojemu mali broj datoteka može sadržavati sve informacije za strujanje bez poteškoća,
- IIS posjeduje korisne alate za analizu, te sposobnost dostave više MSS i HLS sadržaja direktno iz IIS poslužitelja,
- brza adaptacija tijekom HTTP strujanja je ostvarena kada je preporučena upotreba malih dijelova videozapisa prihvaćena,
- skupni format datoteka podržava višestruke podatke koji mogu biti korišteni za pohranu metapodataka o umetanju reklamnog sadržaja, podnapisa.

Nedostatci MSS-a:

- protok podataka kroz mrežu je malo kompleksniji i nedostatak je dodan u trenutku kada je potrebno ubaciti ISS poslužitelj u tok,
- na osobnom računalu potrebna je instalacija zasebnih Silverlight dodataka, [9].

4.2.3. Adobe HTTP Dynamic Streaming (HDS)

Od kada je izašao Adobe Flash Player 10.1, i OSMF 1.0 (engl. *Open Source Media Framework*), pružatelji dostavljanog sadržaja, izrađivači sadržaja i izdavači su

imali opciju da iskoriste snagu HTTP Dynamic Streaming-a, kako bi povećali kvalitetu isporučenoga videosadržaja. HDS je istinska tehnologija za strujanje videozapisa, ali nije ovisna o specijaliziranim poslužiteljima za strujanje ili vlasničkim transportnim protokolima. Alati potrebni kako bi se medijski sadržaj strujao preko HTTP-a su osigurani od strane Adobe-a, [11].

HDS je nasljednik RTMP-a, ali ipak podržava adaptivno strujanje te posjeduje reputaciju visoke kvalitete. HDS je također jedan od najboljih protokola kada je riječ o kašnjenju, koja nije tako mala kao kod RTMP-a, iz razloga što se moraju provesti operacije fragmentacije i enkripcije. Takve karakteristike ga čine popularnima za strujanje sportskih i sličnih događanja, gdje je malo kašnjenje jedan od glavnih prioriteta, [3].

Prednosti HDS-a:

- *flash* klijent je dostupan na raznim uređajima i instaliran je na skoro svakom osobnom računalu na svijetu,
- HDS je dio Flash-a te može koristiti njegovo okruženje i spremno iskoristiti to kao bazu za razvojne programere.

Nedostatci HDS-a:

- HDS spada u skupinu novijih protokola te može imati problema sa stabilnošću više nego njegovi stariji konkurenti,
- ekosustav partnera koji nudi odgovarajuća rješenja je teško limitiran zbog binarnog formata bootstrap⁵ datoteke.

4.2.4. Dynamic Adaptive HTTP Streaming (DASH)

DASH je prvi protokol baziran na HTTP adaptivnom strujanju koji je postao internacionalni standard, razvijen pod pokroviteljstvom MPEG-a. MPEG-DASH je tehnologija koja u principu ujedinjuje ključne karakteristike HDS-a, HLS-a i MSS-a i s još mnogo poboljšanja koja su rezultat razvijanja standarda kroz protekle godine. Standardizacija rješenja s adaptivnim strujanjem je potaknula samopouzdanje svih

⁵ *bootstrap, boot loader* – program koji se pokreće s ROM-a ili RAM-a pri uključivanju računala bez operativnog sustava

koji sudjeluju u industriji, kao rješenje koje može biti prihvaćeno globalno te uz to rješava fragmentaciju koja proizlazi od rješenja dobavljača, [3].

DASH je protokol koji nudi najviše mogućnosti, ali i najkompleksniji od svih navedenih protokola iz razloga što objedinjuje mnoge mogućnosti slične onima u HLS-u i MSS-u, [9].

Prednosti DASH-a:

- baziran je na otvorenome standardu,
- DASH-264 specifikacija ima snažnu potporu industrije koja predviđa dobru interoperabilnost.

Nedostatci DASH-a:

- sveobuhvatljiv je, stoga različite DASH implementacije će najvjerojatnije biti nekompatibilne,
- intelektualna vlasnička prava povezana sa DASH-om nisu u potpunosti jasna. Uobičajeno je da posjednici nužnih patenata sudjeluju u MPEG-ovim procesima za ostale standarde. U slučaju DASH-a, slika nije potpuno razjašnjena, [9].

5. KARAKTERISTIKE NOVOG QUIC PROTOKOLA

QUIC (engl . *Quick UDP Internet Connections*) je eksperimentalni protokol koji obavlja funkcije transportnog sloja razvijen od strane Google-a. Glavni cilj ovog protokola je smanjenje kašnjenja u odnosu na TCP. QUIC se može gledati kao protokol sličan protokolarnom složaju TCP+TLS+HTTP/2 implementiran na UDP-u. Iz razloga što je TCP razvijen u najnižim slojevima računalstva (operativni sustavi, kerneli itd.), raditi preinake na TCP-u je skoro pa nemoguće, jer treba uzeti u obzir koliko bi nadogradnju trebalo napraviti. Kako je QUIC razvijen iznad UDP-a, on nema takva ograničenja i može biti integriran u *end host*⁶ aplikacije, [12].

Trenutno, ugradnja na klijentovoj strani postoji kao dio Chrome-a i Android-a i koristi se u trenutku kada se pristupa poslužiteljskoj aplikaciji koja ju podržava (Google docs i drive, Youtube itd.). Procjene kažu da 88% internetskog prometa koji je stvoren na Android-u te Chrome verziji za stolna računala, koristi QUIC protokol kao bazu i da interakcija sa Google-om rezultira s 5% poboljšanim performansama, ali i sa 30% manje zastajanja zbog učitavanja na videozapisima koji se prenose strujanjem, [12]. U praksi, korištenje QUIC protokola može ponuditi ubrzanje na razini pretraživača, ali glavno područje gdje se poboljšanja trebaju osjetiti, je kod rezidencijalnih korisnika, koji posjeduju brzinu veze od oko 100 Mb/s.

5.1. KLJUČNE PREDNOSTI

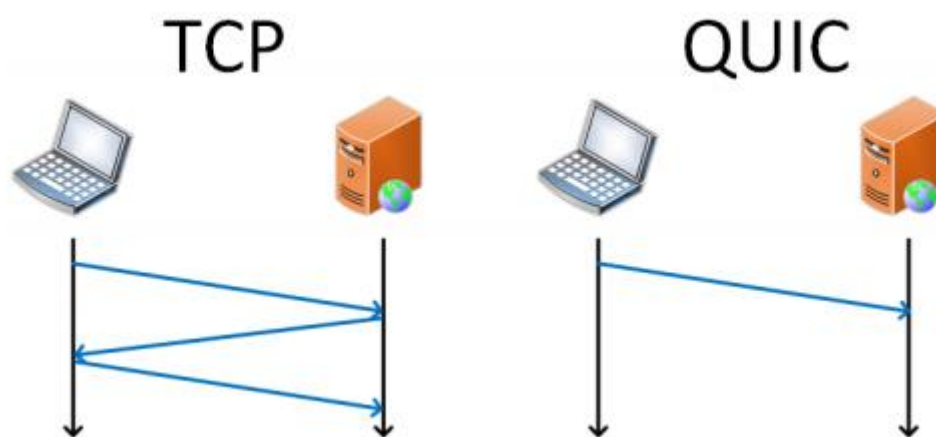
QUIC je funkcijski jednak TCP+TLS+HTTP/2 protokolarnom složaju, ali je razvijen iznad UDP-a. Glavne prednosti u odnosu na navedeni složaj obrazložiti će se u sljedećim potpoglavljima.

5.1.1. Kašnjenje pri uspostavljanju veze

QUIC kombinira transportne i kripto korake, te tako smanjuje potreban broj koraka za uspostavljanje sigurne veze. QUIC veze su najčešće uspostavljene sa 0 koraka (osim u slučaju kada se veza ostvaruje prvi put), što znači da kod većine QUIC veza, podaci mogu biti poslani bez povratne informacije od strane poslužitelja. Za usporedbu, kao što se može vidjeti na slici 6, jedan do tri koraka su potrebna da bi se podaci mogli poslati kod TCP+TLS složaja, [13].

⁶ *end host* – krajnja računala koja su povezana sa računalnom mrežom

Kada se QUIC klijent poveže na QUIC poslužitelj prvi put, on pošalje praznu poruku, zvanu CHLO (engl. *Client Hello*). Poslužitelj tada odgovara sa odbijenicom (engl. *Rejection*), koja posjeduje konfiguraciju poslužitelja i certifikate. Klijent koristi te informacije kako bi poslao još jedan CHLO (sada već može slati aplikacijske podatke), koji je onda prihvaćen od strane poslužitelja i tada svi uzastopno slani podatci budu kriptirani i autentificirani. Kad je poslužitelj poznat klijentu, CHLO je nepotreban, što rezultira sa 0 potrebnih koraka za uspostavu veze između klijenta i poslužitelja, [14].



Slika 6. Uspostava veze kod TCP i QUIC protokola, [14]

5.1.2. Fleksibilna kontrola zagušenja

QUIC posjeduje kontrolu zagušenja koja je uključena nego kod TCP-a i obilniju signalizaciju od TCP-a, što omogućuje QUIC-u da pruži više informacija algoritmima zaduženima za kontrolu zagušenja. Primjer obilnijeg informiranja je da svaki paket, originalni ili ponovno poslani, nosi sa sobom novi sekvencijski broj (engl. *Sequence Number*). To omogućuje QUIC pošiljatelju da razlikuje ACK⁷ za ponovno slanje od ACK-a za originalno slanje, te se tako izbjegava ponovno slanje koje se događa kod TCP-a. QUIC-ovi ACK-i također sa sobom nose kašnjenje koje nastaje između primitka paketa, te njegove potvrde za primitkom, [13].

⁷ ACK (engl. *acknowledgement*) – poruka proslijeđena u komunikacijskim procesima, koja označava potvrdu da je paket na odredišnoj strani primljen

QUIC je otporniji od TCP-a kod ponovnog stvaranja redosljeda (engl. *reordering*), te može zadržati više bajta u prijenosnom linku kada se događa ponovni redosljed ili gubitak. Oba, klijent i server imaju jasniju predodžbu, koje pakete je primatelj zaprimio, [13].

5.1.3. Strujanje i kontrola toka

QUIC posjeduje kontrolu strujanja te kontrolu toka, koja je vrlo slična HTTP/2 kontroli toka. Neophodno je limitirati količinu podataka koju pošiljalac prosljeđuje u bilo kojem trenutku, te isto tako limitirati brzog pošiljalca da sporog primatelja preplavi sa podacima. Ovaj dio će pojasniti načine na koji mehanizmi za kontrolu kod QUIC-a funkcioniraju. Primatelj objavljuje broj okteta koji je spreman zaprimiti pri jednom strujanju ili cijeloj vezi. To dovodi do dvije kontrole:

- kontrola toka za vezu – koja sprječava pošiljalca da pošalje količinu podataka koju primatelj nije u mogućnosti zaprimiti, zbog malog kapaciteta spremnika,
- kontrolu toka za strujanje – koja sprječava jednom strujanju, zauzimanje cijelog kapaciteta spremnika za vezu, [15].

Kako su podatci poslani, zaprimljeni i dostavljeni putem jednoga strujanja, primatelj šalje WINDOW_UPDATE okvire koji povećavaju objavljeni limit za to strujanje, te tako dozvoljava pošiljalcu slanje većeg broja podataka, [13].

5.1.4. Multipleksiranje

TCP koristi TCP portove i IP adrese (na oba kraja) kako bi identificirao vezu, kada višestazni TCP nije korišten. Tako nije moguće za klijenta, da komunicira sa poslužiteljem preko više portova, preko jedne veze. QUIC koristi 64 bitne identifikatore za vezu (koji su nasumice odabrani od strane klijenta). U tim ostvarenim vezama, višestruka strujanja su korištena za transport segmenata. Ovo omogućava klijentu uspostavljanje mobilne veze diljem IP adresa i UDP portova. Također je moguće korištenje višestrukih portova za aplikaciju, ali u tom slučaju aplikacija mora slušati sve UDP portove, jer QUIC koristi UDP kao protokol koji obavlja funkcije transportnog sloja, [14].

Identifikatori veze također omogućavaju migraciju veze. Veza može opstati iako je IP adresa na jednom kraju promijenjena. Podrška višestrukih strujanja u jednostrukoj vezi također adresira *Head-of-line*⁸ blokiranje tako što se šalju neovisni podatci preko različitih strujanja. Sva strujanja se mogu identificirati s identifikatorima strujanja (engl. *Stream ID*) i mogu biti ostvarena od strane klijenta ili poslužitelja. Kako bi se izbjegli sudari, identifikatori strujanja moraju biti parni, kada klijent inicira strujanje i neparni kada je strujanje inicirano od strane poslužitelja. Svaki sudionik mora povećati identifikator strujanja za nova strujanja, [14].

5.1.5. Autentifikacija i enkripcija

TCP zaglavlja (i korisnički podaci, sve dok se ne koristi niti jedan drugi protokol) nisu kriptirana niti autentificirana. QUIC paketi su uvijek kriptirani (osim javnog zaglavlja) i autentificirani (uključujući javno zaglavlje) nakon ostvarene veze. Ovo također uključuje zaštitu za podvalu IP adresa.

5.1.6. Forward Error Correction (FEC)

Kako bi se oporavili izgubljeni paketi, bez potrebnog čekanja za ponovno slanje, QUIC trenutno koristi jednostavnu XOR-baziranu FEC shemu. FEC paket sadrži paritet paketa u FEC grupi. Ako je jedan paket u grupi izgubljen, sadržaj tog paketa može biti oporavljen od strane FEC paketa i ostalih paketa u grupi. Pošiljatelj može odabrati želi li poslati FEC pakete kako bi se optimizirali posebni slučajevi, [13].

5.1.7. Migracija veze

QUIC omogućuje vezi da preživi promjene na krajnjim adresama, kao što su one koje nastaju migracijom krajnjih točaka na novu mrežu. Krajnja točka ne smije inicirati migraciju veze prije nego što je ona ostvarena. Dizajn QUIC-a se oslanja na stabilne adrese krajnjih točaka, koje takve i ostaju tijekom cijelog vremena uspostavljanja veze, [15].

QUIC veze su identificirane 64-bitnim identifikatorom veze, nasumično određen od strane klijenta. QUIC može preživjeti promjenu IP adrese, pa tako identifikator veze ostaje isti tijekom cijelog vremena trajanja migracije. QUIC također

⁸ *Head-of-line blokiranje* – fenomen koji se događa kada se niz paketa zaustavi zbog prvog paketa

omogućava automatsku kriptografsku potvrdu klijenta koji migrira, kako on nastavlja koristiti isti sesijski ključ za enkripciju i deskripciju paketa, [13].

5.2. PAKETIZACIJA I POUZDANOST

Pošiljalatelj usnopljuje jedan ili više okvira u jednom QUIC paketu. Pošiljalatelj bi trebao minimizirati prijenosni kapacitet po paketu, te računalne troškove po paketu, tako što bi trebao usnopljavati što je moguće više okvira u jednom QUIC paketu. Pošiljalatelj možda bude čekao jedan kratak period vremena da stvori snop od više okvira, prije nego se paket pošalje, kako bi se izbjeglo slanje velikog broja malih paketa. Može koristiti poznavanje aplikacija te njihovog ponašanja kako bi se utvrdilo treba li čekati, te koliko dugo bi trebalo čekati. Taj period čekanja je predan na odluku prilikom implementacije, te bi se ta odluka trebala oprezno razmatrati, iz razloga što svako kašnjenje može povećati vidljivo kašnjenje pri upotrebi aplikacije, [15].

5.2.1. Procesiranje paketa i ACK

ACK poruka ne smije biti poslana sve dok zaštita paketa nije uspješno uklonjena i sve dok se svi okviri u paketu ne procesiraju. Za okvire kojima se omogućuje strujanje, može se dogoditi da podatci budu u redu za čekanje, te pripremani za njihovo primanje od strane aplikacijskog protokola, ali ne zahtjeva se njihova dostava i upotreba. Jednom kada paket biva procesiran, primatelj daje potvrdu tako što šalje jednu ili više ACK poruka koje posjeduju broj paketa koji su zaprimljeni. Kako bi se spriječila beskonačna petlja, krajnja točka ne smije slati ACK poruku kao odgovor paketu koji je ispunjen samo ACK porukom ili *padding*⁹ okvirom, čak iako postoje praznine koje prethode primljenom paketu, [15].

Iako *padding* okviri ne traže ACK poruke od strane primatelja, oni se smatraju dijelom toka, zbog kontrole zagušenja. Slanjem samo *padding* okvira, može izazvati ograničenje kod pošiljalatelja, od strane kontrole zagušenja, bez ACK poruka koje dolaze od strane primatelja. Zbog toga, pošiljalatelj bi morao osigurati da su ostali okviri poslani kao dodatak *padding* okvirima, kako bi dobio ACK poruku od primatelja, [15].

⁹ *padding* – proces pri paketizaciji s kojim se povećava duljina paketa, kako bi paket bio valjan

5.2.2. Ponovno slanje informacija

Paketi QUIC-a koji su izgubljeni ne prenose se u cijelosti. Isto vrijedi i za okvire koji su sadržani unutar izgubljenih paketa. Umjesto toga, informacije koje se mogu prenijeti u okvirima, ponovno se šalju u novim okvirima po potrebi. Novi okviri i paketi se koriste za prijenos podataka za koje je utvrđeno da su izgubljeni. Općenito, podatci se ponovno šalju, kada se utvrdi da je paket koji sadrži podatke izgubljen, a slanje prestaje kada paket koji prenosi informacije, bude potvrđen ACK porukom, [15].

5.2.3. Veličina paketa

QUIC paket uključuje njegovo zaglavlje te provjeru integriteta, ali ne uključuje IP ili UDP zaglavlje. Klijent mora osigurati da prvi inicijalni paket koji se pošalje, bude poslan u UDP datagramu, te da se sastoji od najmanje 1200 okteta. Kako bi se zadovoljio navedeni kriterij, može se nadopuniti (engl. *padding*) paket ili nadodati 0-RTT (engl. *zero round trip time*) paket u isti datagram. Slanje UDP datagrama ovakve veličine osigurava da će mrežni put podržavati prihvatljiv MTU (engl. *Maximum Transmission Unit*), te pomoći smanjiti amplitudu povećanja napada, izazvanih poslužiteljevima odgovorima prema neverificiranim adresama klijenata, [15].

5.2.4. Path Maximum Transmission Unit (PMTU)

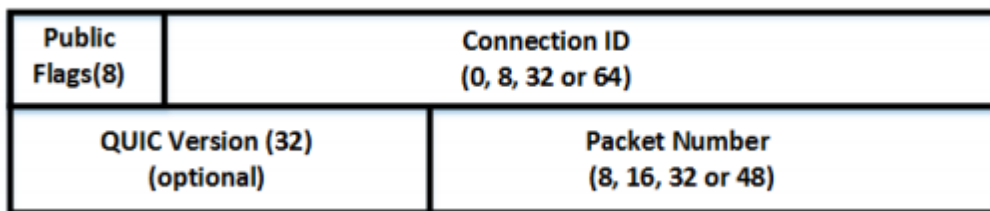
PMTU je maksimalna veličina cijeloga IP zaglavlja, UDP zaglavlja i UDP-ovih korisničkih informacija. Korisničke informacije koje se prenose UDP-om sadrže QUIC zaglavlje paketa, zaštićene korisničke informacije i autentifikacijska polja. Svi QUIC paketi bi trebali biti veličine, koja bi odgovarala procijenjenom PMTU-u, kako bi se izbjegla fragmentacija ili ispuštanje paketa. Kako bi se optimizirala efikasnost prijenosnog kapaciteta, krajnje točke bi trebale koristiti PLPMTUD (engl. *Packetization Layer PMTU Discovery*). Nedostatak ovog mehanizma je taj, da QUIC krajnje točke ne bi trebale slati pakete veće od 1280 okteta. QUIC krajnje točke koje implementiraju bilo kakav PMTU *discovery*, bi trebale održavati procjenu za svaku kombinaciju lokalnih i udaljenih IP adresa, [15].

Svako spajanje lokalne i udaljene IP adrese bi moglo dovesti do različitih MTU-a na putu. QUIC ovisi o mrežnom putu koji podržava MTU sa barem 1280

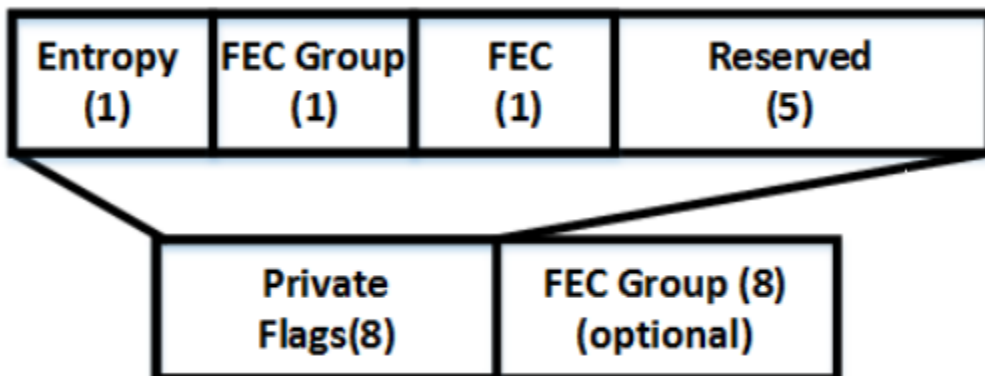
okteta. To je minimum za IPv6 te također podržan od strane najmodernijih IPv4 mreža. Krajnja točka ne smije smanjiti MTU ispod navedenoga broja, čak i ako primi signal koji daje do znanja da postoji manje ograničenje. Ako krajnja točka QUIC-a shvati da je PMTU između bilo kojega para lokalne i udaljene IP adrese pao ispod 1280 okteta, mora istoga trenutka prekinuti slanje QUIC paketa tim putem. Navedeni slučaj bi mogao rezultirati završetkom postojeće veze, ako alternativna ruta nije pronađena, [15].

5.2.5. Tipovi paketa i format zaglavlja

QUIC razlikuje dva tipa paketa: regularne i specijalne pakete. Specijalni paketi mogu biti paketi za pregovaranje (engl. *Version Negotiation Packets*) ili javni paketi za resetiranje (engl. *Public Reset Packets*). Regularni paketi su podijeljeni u okvirne pakete (engl. *Frame Packets*) i FEC pakete (engl. *FEC Packets*). Paketi posjeduju dva tipa zaglavlja: nekriptirano javno zaglavlje (engl. *Public Header*), (slika 7.) i kriptirano privatno zaglavlje (engl. *Private Header*), (slika 8.), [14].



Slika 7. QUIC javno zaglavlje (brojevi u bitima), [14]



Slika 8. QUIC privatno zaglavlje (brojevi u bitima), [14]

5.3. QUIC STRUJANJE

Strujanja su neovisan niz dvosmjernih podataka u okvirima za strujanje. Strujanja mogu biti izazvana od strane poslužitelja ili klijenta, mogu istovremeno slati podatke koji se preklapaju s drugim strujanjima te se mogu otkazati. Stvaranje strujanja je napravljeno jednostavno, šaljući okvire za strujanje (engl. *Stream Frame*) za zadano strujanje. Kako bi se izbjegao sudar među identifikatorima strujanja, već je ranije navedeno da identifikator strujanja mora biti paran, ako ga inicira poslužitelj, te neparan ako je iniciran od strane klijenta. *Stream 0* nije valjan identifikator, dok je *Stream 1* rezerviran za ostvarivanje sigurne veze (engl. *cryptographic handshake*), te bi to trebalo biti prvo strujanje inicirano od strane klijenta, [13].

Kada se koristi HTTP/2 preko QUIC-a, *Stream 3* je rezerviran za prijenos zaglavlja od svih drugih strujanja, tako osiguravajući pouzdanu i ispravnu dostavu te obradu zaglavlja. Identifikatori strujanja sa svake strane veze se moraju monotono povećavati, kako se nova strujanja stvaraju. Na primjer, *Stream 2* može biti stvoren nakon *Stream-a 3*, ali *Stream 7* ne smije biti stvoren nakon *Stream-a 9*. *Peer*¹⁰ može zaprimiti strujanja koja nisu poredana kako bi trebala biti. Ako krajnja točka ne želi zaprimiti strujanje, može istoga trenutka odgovoriti sa *RTS_STREAM* okvirom. U slučaju da krajnja točka, koja je inicirala strujanje, pošalje podatke prije odobravanja strujanja od druge točke, ti podaci moraju biti zanemareni, [13].

Jednom kada je strujanje ostvareno, može biti upotrijebljeno kako bi se slali ili zaprimali podaci. To znači da niz okvira strujanja mogu biti poslani QUIC krajnjoj točki tijekom strujanja, sve dok strujanje nije završeno. Postoje tri načina kojima strujanje može biti završeno:

- Normalno završavanje (engl. *Normal termination*): Kako su strujanja dvosmjerna, mogu biti „poluzatvorena“ i „zatvorena“. Kada jedna strana strujanja pošalje okvir gdje je FIN bit postavljen kao istinit, strujanje se smatra kao „poluzatvoreno“ u tom smjeru. FIN daje do znanja da neće postojati podaci u budućnosti koji će biti poslani od strane pošiljatelja FIN-a. Kada obje krajnje točke pošalju i zaprime FIN, strujanje se tretira kao „zatvoreno“. FIN bi trebao biti poslan sa zadnjim korisničkim podacima

¹⁰ *peer-to-peer* – koncept umrežavanja bez poslužitelja, svako računalo inteligentna radna stanica, svi su u jednakom odnosu

strujanja, a može biti poslan i u praznom okviru koji prati zadnje podatke na tom strujanju.

- Isprekidano završavanje (engl. *Abrupt termination*): Klijent ili poslužitelj mogu poslati RST_STREAM okvir za strujanje u bilo kojemu trenutku. Navedeni okvir sadrži pogrešan kod koji ukazuje razlog zbog kojeg je došlo do prekida. Kada je navedeni okvir poslan od strane pokretača strujanja, onda on ukazuje na prekid kako bi se strujanje okončalo, te kako se neće slati dodatni podaci u tom strujanju. Kada je RST_STREAM okvir poslan od strane primatelja, pošiljatelj, po primitku, bi trebao prestati slati bilo kakve podatke prilikom tog strujanja.
- Strujanja su također završena kada je veza okončana, [13].

5.4. ZAVRŠETAK VEZE

Veza bi trebala ostati otvorena sve dok ne postane besposlena određeni unaprijed dogovoreni period vremena. Kada poslužitelj odluči okončati besposlenu vezu, ne bi trebao obavijestiti klijenta kako ne bi „probudio“ mobilni uređaj. QUIC veza, jednom kada je ostvarena može biti okončana na dva načina:

- Eksplicitno isključivanje (engl. *Explicit Shutdown*): Krajnja točka šalje CONNECTION_CLOSE okvir korisniku koji je inicirao završetak veze. Krajnja točka možda pošalje GOAWAY okvir prema korisniku prije CONNECTION_CLOSE okvira, kako bi se dalo do znanja da će veza uskoro biti prekinuta. Kada se pošalje GOAWAY okvir, signalizira se korisniku da će se sva aktivna strujanja obrađivati, ali pošiljatelj navedenoga okvira neće inicirati nikakva dodatna strujanja te neće prihvatiti nikakva nadolazeća strujanja.
- Implicitno isključivanje (engl. *Implicit Shutdown*): Zadani prekid u stanju mirovanja za vezu u QUIC-u iznosi 30 sekundi, maksimalni je 10 minuta. U slučaju da nema mrežne aktivnosti u iznosu vremena mirovanja, veza je završena. Po zadanim postavkama, CONNECTION_CLOSE okvir će biti poslan. Tihi način završetka veze može biti omogućen kada je preskupo eksplicitno završavanje veze, kao što je slučaj kod mobilnih mreža koje moraju „probuditi“ uređaj, [13].

Krajnja točka može također poslati PUBLIC_RESET paket u bilo kojem trenutku da iznenada završi aktivnu vezu. PUBLIC_RESET je ekvivalent TCP-ovom RST-u, [13].

6. UTJECAJ NA KVALITETU USLUGE

Internet pretraživači su postali preferencijalna platforma preko koje se pristupa velikom broju usluga, uključujući pretraživanje, zabavu, posao, društvene mreže, osobna komunikacija itd. U važnim vremenima razvijanja, od HTTP/1 protokola do HTTP/2, SPDY¹¹ i QUIC-a, imati pouzdane puteve u kojima se mogu usporediti performanse protokola, postaje važan zadatak, prije nego masivno razvijanje počne. Brojna istraživanja ukazuju na važnost kašnjenja, te direktnom povezanošću s vrijednošću pri poslovanju, [16].

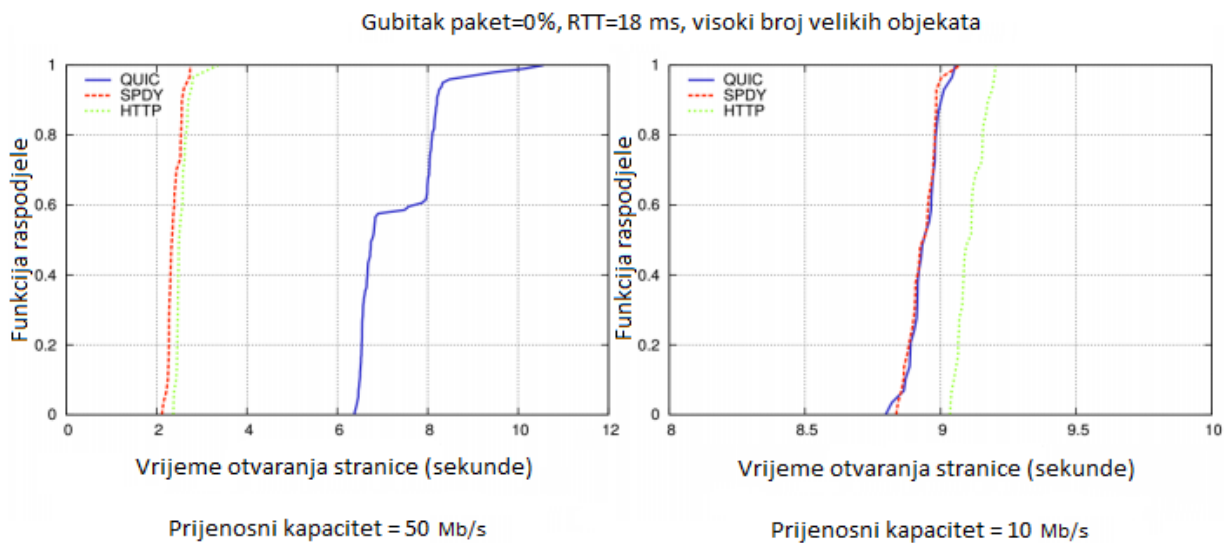
Na primjer, tvrtke Amazon i Google su prijavile gubitak u rasponu od 0.6 do 1.2%, kada se vrijeme učitavanja web stranice povećalo 0,4 do 1 sekundu, a tvrtka Shopzilla prijavljuje povećanje prihoda od 12% kada radi podešavanje web stranice, te smanjuje vrijeme učitavanja za čak 5 sekundi. Skrivena povezanost između ovih faktora ja naravno, utjecaj kašnjenja na kvalitetu usluge web korisnika, [16].

Grupa ljudi sa odjela za telekomunikacije i medijske informacije, na sveučilištu tehnologije i ekonomije u Budimpešti, testiralo je QUIC sa HTTP i SPDY protokolom, [14]. Njihovo okruženje u kojemu je testirano, te rezultati, prikazani su u ovome poglavlju. Grupa je napravila višestruke testove, ali prikazati će se samo najinteresantniji. Uobičajeno prijenosno računalo sa Google Chrome pretraživačem je korišteno u svrhu ovoga testiranja (na strani klijenta). Na strani poslužitelja, korištene su četiri jednostavne stranice. U ovome scenariju, stranice su sadržavale vrlo male (400 B – 8kB) ili vrlo velike (128 kB) objekte. Za svaki scenarij, bila je stranica sa malo (5) ili stranica sa puno (50) navedenih objekata. U usporedbi sa Google-ovim testiranjima, u ovome testiranju nisu istraživani utjecaji na omogućavanje i onemogućavanje različitih mehanizama, [14].

U jednome od testova, pokazalo se da QUIC radi lošije u scenariju gdje je prijenosni kapacitet 50 Mb/s, nizak RTT (18 ms), razina gubitka paketa 0% te 50 objekata veličine 128 kB. Kada je TCP korišten (s i bez SPDY-a) za HTTP vezu, prosječno trajanje otvaranja stranice iznosilo je 2 sekunde, dok je kod QUIC-a iznosilo 7 sekundi. Kada se prijenosni kapacitet smanjio na 10 Mb/s, performanse

¹¹ SPDY – protokol razvijen od strane Google-a, za transport web sadržaja

QUIC-a su opet postale usporedive sa ostalima i prosječno vrijeme otvaranja stranice je iznosilo 9 sekundi (slika 9), [14].

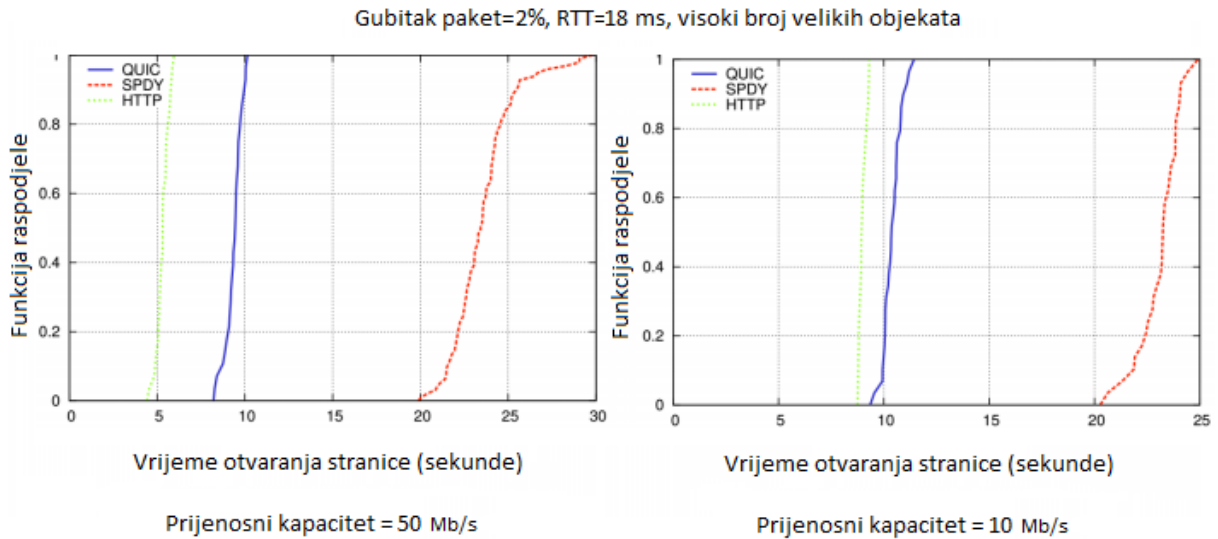


Slika 9. Usporedba QUIC-a, HTTP-a i SPDY-a pod navedenim okolnostima, [14]

Razlog slabih performansi QUIC-a je packet pacing¹². Taj mehanizam pokušava smanjiti ponovno slanje paketa, tako što šalje pakete, sporije od svoje maksimalne mogućnosti. U ovome scenariju, gubitak paketa iznosi 0%, pa navedeni proces pokušava izbjeći gubitak, koji ne bi bio prisutan ni u slučaju da se prijenos ostvari punom brzinom. QUIC podcjenjuje maksimalni prijenosni kapacitet te šalje premalo podataka u jedinici vremena. Kada se koristi manji prijenosni kapacitet (10 Mb/s), takvo ponašanje nije uočljivo, [14].

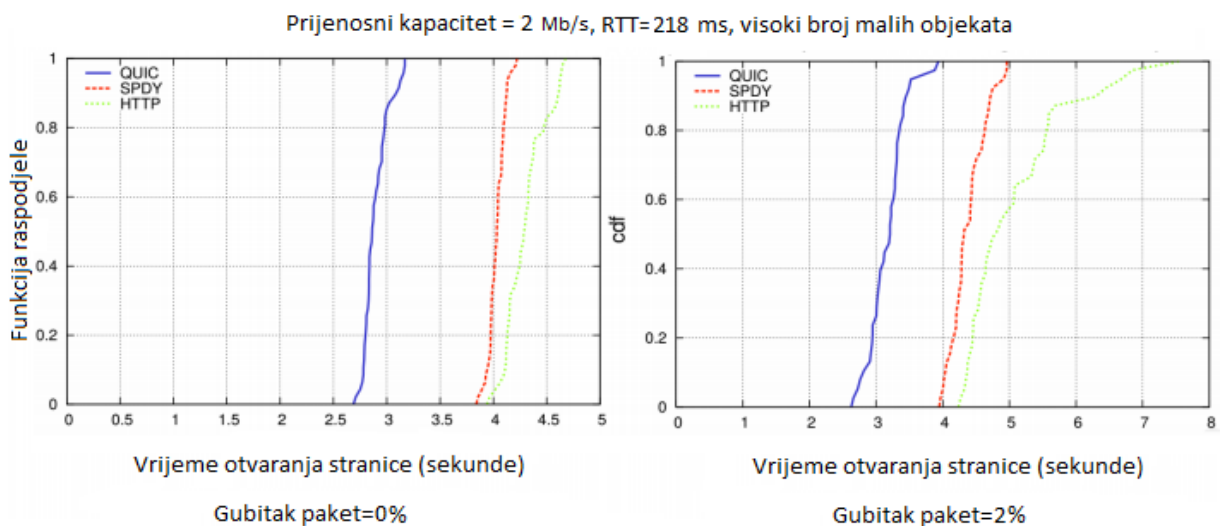
Potpuno drugačije je ponašanje, kada je razina gubitka paketa promijenjena. Slika 10 ukazuje na utjecaj kada se gubitak postavi na 2%. HTTP i dalje djeluje bolje od QUIC-a, ali razlika je smanjena. U usporedbi, SPDY djeluje vrlo loše. Razlog za to je proces pod imenom *Head-of-line* blokiranje, koji je objašnjen u jednom od prijašnjih poglavlja.

¹² packet pacing – skup tehnika, kojima se uzorkuje prijenos paketa, kako bi proces prijensa bio konstantniji



Slika 10. Usporedba QUIC-a, HTTP-a i SPDY-a pod navedenim okolnostima, [14]

Google tvrdi da QUIC radi puno bolje od ostalih protokola, kada se koristi za spore veze sa velikim RTT-om, posebno kada se zahtijeva veliki broj malih objekata. U sljedećem scenariju korišten je prijenosni kapacitet od 2 Mb/s, RTT od 218 ms i 50 objekata između 400B i 8kB. Rezultati dobiveni (slika 11) potvrđuju Google-ove tvrdnje. QUIC radi bolje u oba slučaja (0% i 2% gubitak paketa). Prednosti su većinom dobivene zbog mehanizma multipleksiranja QUIC-a. Također se došlo do zaključka da 0 RTT veza ima pozitivan utjecaj na učitavanje stranice, [14].



Slika 11. Usporedba QUIC-a, HTTP-a i SPDY-a pod navedenim okolnostima, [14]

7. ZAKLJUČAK

QUIC je transportni protokol, razvijen od strane Google-a. Navedeni protokol je još u fazi izrade, pa svi mehanizmi koji se spominju nisu implementirani. QUIC koristi UDP kao protokol četvrtog sloja, ali neki mehanizmi su preuzeti i od strane TCP-a. QUIC posjeduje neke funkcionalnosti koje TCP nema, što je i logično jer ne bi imalo smisla razvijati protokol sa istim funkcionalnostima kao neki koji već postoji. Usporedbe radi, QUIC može posjedovati kriptiranu vezu sa 0RTT. Google tvrdi da QUIC djeluje, u najgorem slučaju, isto kao TCP, što istraživanja koja su prezentirana u ovome radu pokazuju da nije istina.

Prednosti QUIC-a dolaze prvenstveno zbog multipleksiranja te 0RTT veze. Google je već ranije implementirao QUIC na neke svoje usluge, ali treba se uzeti u obzir da je navedeni protokol još u fazi razvijanja. U slučaju kada Google bude htio pozicionirati QUIC, te ga implementirati na više poslužitelja, protokol će se morati poboljšati te će svi navedeni mehanizmi morati biti razvijeni i implementirani. Protokol je superioran u slučaju sa velikim brojem malih objekata, velikim RTT-om te malim raspoloživim prijenosnim kapacitetom, ali za široku upotrebu će trebati dodatna poboljšanja. QUIC je protokol velikog potencijala, ali pitanje je li hoće li uspjeti funkcionirati paralelno sa TCP-om ili ga u potpunosti zamijeniti u skoroj budućnosti.

LITERATURA

- [1] **Rao, A., Barakat, C., Legout, A., Towsley, D.** *Network Characteristics of Video Streaming Traffic*. Studeni 2011. **URL:**<https://hal.inria.fr/inria-00638063/document> (pristupljeno: kolovoz 2018.)
- [2] **Mrvelj, Š., Matulin, M.** *Autorizirana predavanja, Tehnologije TK Prometa 1*. Veljača 2018. **URL:**https://moodle.srce.hr/2017-2018/pluginfile.php/1210640/mod_resource/content/5/Tehnologija%20telekomunikacijskog%20prometa%203%20predavanje%20za%20objavu.pdf (pristupljeno: kolovoz 2018.)
- [3] **Wilbert, M.** *Video Streaming Protocol*. Studeni 2017. **URL:**<https://www.dacast.com/blog/video-streaming-protocol/> (pristupljeno: kolovoz 2018.)
- [4] **Santos-González, I., Rivero-García, A., Molina-Gil, J., Caballero-Gil, P.** *Implementation and Analysis of Real-Time Streaming Protocols*. Travanj 2017. **URL:**<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5424723/> (pristupljeno: kolovoz 2018.)
- [5] *Streaming Protocols - A brief History*. Prosinac 2016. **URL:**<https://dveo.com/pdf/Streaming-Protocols-A-Brief-History.pdf> (pristupljeno: kolovoz 2018.)
- [6] *Streaming Stored Audio and Video*. **URL:**https://www.net.t-labs.tu-berlin.de/teaching/computer_networking/06.02.html (pristupljeno: kolovoz 2018.)
- [7] **Čandrić, G.** *RTMP – Real Time Messaging Protocol Explained*. Svibanj 2014. **URL:**<https://www.globaldots.com/rtmp-real-time-messaging-protocol-explained-2/> (pristupljeno: kolovoz 2018.)
- [8] *Adaptive streaming*. **URL:**<https://bitmovin.com/adaptive-streaming/> (pristupljeno: kolovoz 2018.)
- [9] **Yuval, F.** *AN OVERVIEW OF HTTP ADAPTIVE STREAMING PROTOCOLS FOR*. 2014. **URL:**<https://www.nctatechnicalpapers.com/Paper/2014/2014-an-overview-of-http-adaptive-streaming-protocols-for-tv-everywhere-delivery> (pristupljeno: kolovoz 2018.)
- [10] **Mueller, C.** *Microsoft Smooth Streaming*. Svibanj 2015. **URL:**<https://bitmovin.com/microsoft-smooth-streaming-mss/> (pristupljeno: kolovoz 2018.)
- [11] *An Introduction to Streaming Media*. **URL:**<http://realeyes.com/blog/2011/07/http-dynamic-streaming-part-1-an-introduction/> (pristupljeno: kolovoz 2018.)
- [12] *QUIC (Quick UDP Internet Connections)*. **URL:**<https://fasterdata.es.net/data-transfer-tools/quic-quick-udp-internet-connections/> (pristupljeno: kolovoz 2018.)

- [13] **Hamilton, R., Iyengar, J., Swett, I., Wilk, A.** *QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2*. Srpanj 2016. **URL:**<https://tools.ietf.org/html/draft-tsvwg-quick-protocol-02#page-5> (pristupljeno: kolovoz 2018.)
- [14] **Gratzer, F.** *QUIC - Quick UDP Internet Connections*. Rujan 2016. **URL:**https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2016-09-1/NET-2016-09-1_06.pdf. (pristupljeno: kolovoz 2018.)
- [15] **Iyengar, J., Thomson, M.** *QUIC: A UDP-Based Multiplexed and Secure Transport*. Travanj 2018. **URL:**<https://quicwg.org/base-drafts/draft-ietf-quick-transport.html#rfc.section.10>. (pristupljeno: kolovoz 2018.)
- [16] **Bocchi, E., De Cicco, L., Rossi, D.** *Measuring the Quality of Experience of Web users*. **URL:**<https://perso.telecom-paristech.fr/drossi/paper/rossi16internet-qoe.pdf>. (pristupljeno: kolovoz 2018.)
- [17] **Jarboe, G.** *Video will account for 79% of global traffic by 2020*. Siječanj 2017. **URL:**<http://tubularinsights.com/video-79-percent-internet-traffic-2020/> (pristupljeno: rujan 2018.)

POPIS KRATICA

QUIC	Quick Udp Internet Connections
UDP	User Datagram Protocol
OSI	Open System Interconnection
TCP	Transmission Control Protocol
HTTP	Hyper Text Transfer Protocol
VNI	Visual Networking Index
HTML	Hypertext Markup Language
CBR	Constant Bit Rate
VBR	Variabile Bit Rate
ABR	Adaptive Bit Rate
RTSP	Real Time Streaming Protocol
RTP	Real Time-Transfer Protocol
RTMP	Real Time Messaging Protocol
FLV	Flash Video
MP4	Mpeg-4
MP3	Mpeg-1 Audio Layer 3
AAC	Advanced Audio Coding
QOS	Quality Of Service
WWW	World Wide Web
HLS	HTTP Live Streaming
URL	Uniform Resource Locator
MSS	Microsoft Silverlight Smooth Streaming
IIS	Internet Information Service
NBC	National Broadcasting Company
HDS	HTTP Dynamic Streaming
OSMF	Open Source Media Framework
DASH	Dynamic Adaptive HTTP Streaming
CHLO	Client Hello
ACK	Acknowledgement

IP	Internet Protocol
FEC	Forward Error Correction
RTT	Round Trip Time
MTU	Maximum Transmission Unit
PMTU	Path Maximum Transmission Unit
PLPMTUD	Packetization Layer PMTU Discovery

POPIS SLIKA

Slika 1. Interakcija između klijenta i poslužitelja korištenjem RTSP-a.....	7
Slika 2. RTP kao podsloj transportnoga sloja.....	8
Slika 3. Put videozapisa od poslužitelja do korisnika progresivnim strujanjem	12
Slika 4. Adaptivno strujanje, različiti videozapisi za različite zaslone	13
Slika 5. Adaptivno strujanje, prilagođavanje na različite brzine veze	13
Slika 6. Uspostava veze kod TCP i QUIC protokola	20
Slika 7. QUIC javno zaglavlje (brojevi u bitima).....	25
Slika 8. QUIC privatno zaglavlje (brojevi u bitima).....	25
Slika 9. Usporedba QUIC-a, HTTP-a i SPDY-a pod navedenim okolnostima	30
Slika 10. Usporedba QUIC-a, HTTP-a i SPDY-a pod navedenim okolnostima	31
Slika 11. Usporedba QUIC-a, HTTP-a i SPDY-a pod navedenim okolnostima	31

POPIS TABLICA

Tablica 1. Usporedba <i>push</i> i <i>pull-based</i> protokola za strujanje videosadržaja.....	6
---	---



Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj završni rad

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu završnog rada

pod naslovom **Analiza značajki protokola QUIC transportnog sloja**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 7.9.2018

Luka Boželić

(potpis)