Afan Čečo
Fra Grge Martića 12
72 000 Zenica
Tel: 061/756-041

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET U SARAJEVU
VIJEĆE DOKTORSKOG STUDIJA

PREDMET: **Prijava prijedloga teme doktorske disertacije (projekta)**

Poštovani,

Obzirom da sam na drugoj godini doktorskog studija Elektrotehničkog fakulteta u Sarajevu, Odsjek za računarstvo i informatiku, molim Vas da izvršite ocjenu podobnosti predložene teme za doktorski rad, odnosno da zakažete termin za odbranu projekta doktorske disertacije, te da imenujete savjetnika, odnosno mentora ili supervizora.

Kao temu doktorskog rada predlažem:
"Prijedlog metode za poboljšanje kontrole zagušenja na Internet mreži"

Obzirom na konsultacije prilikom odabira teme i područje istraživanja koje tema obuhvata, kao savjetnika predlažem doc. dr. Sašu Mrdovića, nastavnika na Elektrotehničkom fakultetu u Sarajevu.

Uz prijavu prilažem:
- Prijedlog teme doktorskog rada
- Curriculum Vitae
- Spisak i fotokopije objavljenih naučnih radova

U očekivanju Vašeg pozitivnog odgovora, srdačno Vas pozdravljam.

S poštovanjem,

Sarajevo, 30. 06. 2014. g. mr. Afan Čečo, dipl. el. ing.

Afan Čečo
Fra Grge Martića 12
72 000 Zenica

Univerzitet u Sarajevu
Elektrotehnički fakultet
Odsjek za računarstvo i informatiku

Prijedlog teme doktorskog rada

Tema rada:

„Prijedlog metode za poboljšanje kontrole zagušenja na Internet mreži“

Naslov teme na engleskom jeziku glasi:

„Proposal of method for improving congestion control over Internet network“

1. Obrazloženje teme

Internet je najveća računarska mreža na svijetu, odnosno „mreža svih mreža“. TCP/IP protokol je standard Interneta. IP protokol omogućava rutiranje paketa između mreža. Međutim, IP ne garantuje da će paketi podataka biti isporučeni. TCP protokol je odgovoran za pouzdan transport i regulisanje toka podataka od izvorišta do odredišta. Osim na nivou TCP protokola, mogući su mehanizmi za pouzdanost transporta i na drugim nivoima.

Problem zagušenja na Internet mreži je sveprisutan i stalno se mora boriti protiv njega. Ogleda se u tome da svi paketi podataka koji se šalju, ne mogu biti isporučeni, kako zbog ograničenosti linkova, tako i zbog ograničenosti procesnih čvorova i njihovih memorijskih kapaciteta. Kod podatkovnih mreža, odnosno u teoriji redova čekanja, se pod zagušenjem smatra stanje kada se na nekom linku ili čvoru nalazi toliko podataka da kvalitet usluge prenosa podataka znatno opada. Obično se tada dešava kašnjenje u isporuci paketa podataka, gubljenje paketa ili blokiranje uspostave novih konekcija. Kada je stanje mreže takvo da postoji niska propusnost smatra se da je nastupio kolaps usljed zagušenja. Kolaps usljed zagušenja je identificiran kao problem i opisan u [1], iz 1984. godine. Prvi put je zabilježen u oktobru 1986. godine, kada je NSFnet backbone spao u pogledu kapaciteta sa 32 kb/s na 40 b/s. To se nastavilo dešavati do 1988. godine kada su čvorovi implementirali Van Jacobson-ovu kontrolu zagušenja [2]. Mehanizmi kontrole zagušenja omogućavaju da se izbjegnu problemi koji mogu nastati prilikom transmisije paketa podataka.

U ovom radu bio bi načinjen pregled mehanizama kontrole zagušenja, kako na nivou TCP protokola, tako i na drugim nivoima (npr. na nivou IP protokola). Detaljno bi bili objašnjeni mehanizmi kontrole zagušenja kod TCP protokola, naročito najviše korištena rješenja. U radu bi se i praktično vidjela primjena ovih mehanizama, te sagledale sve prednosti njihovog korištenja. Bila bi predstavljena moguća poboljšanja mehanizama kontrole zagušenja kod TCP protokola. Rad bi definisao problem zagušenja na Internet mreži, analizirao dosadašnja rješenja i dao bi originalni doprinos u vidu nove metode za poboljšanje kontrole zagušenja na Internet mreži.

U sklopu praktičnog dijela rada bile bi urađene simulacije koje prikazuje različite, prethodno spomenute, mehanizme kontrole zagušenja, kao i njihovo međusobno poređenje. U samim simulacijama, bila bi prikazana moguća poboljšanja kontrole zagušenja na Internet mreži. Naročita pažnja bi bila posvećena poboljšanju mehanizama kontrole zagušenja na nivou TCP protokola, jer to predstavlja centralno mjesto za kontrolu, odnosno izbjegavanje zagušenja u Internet mreži.

2. Stanje u oblasti kojoj tema pripada

Do sada su uloženi veliki napori na razvoju različitih mehanizama kontrole zagušenja na Internet mreži. Najveći napori su uloženi na transportnom sloju TCP/IP (ili OSI) modela, na kojem se nalazi TCP protokol, jer je njegov primarni zadatak da bude upravljački protokol, odnosno da osigura pouzdanost prilikom prenosa podataka, što uključuje i kontrolu zagušenja. Postoji veći broj različitih verzija TCP protokola, počevši od Tahoe TCP-a, preko Reno TCP-a, New Reno TCP, zatim SACK TCP-a, itd. Osim na transportnom sloju, postoje i mehanizmi kontrole zagušenja na mrežnom sloju, na kojem se nalazi IP protokol (kao što su ECN ili RED).

2.1. Kontrola zagušenja na mrežnom sloju

Metode za kontrolu zagušenja na mrežnom sloju podrazumijevaju učešće mrežnih čvorova (rutera) koji rade na ovom sloju, za razliku od onih na transportnom sloju, gdje se ne očekuje nikakva podrška od rutera.

2.1.1. *Explicit Congestion Notification* (ECN)

Eksplicitno obavještenje o zagušenju (eng. Explicit Congestion Notification, ECN) je proširenje Internet protokola (IP) i definisano je u [3]. ECN omogućava *end-to-end* notifikaciju mrežnog zagušenja, bez odbacivanja paketa. Radi se o opcionalnoj mogućnosti i primjenjuje se samo kada oba kraja signaliziraju da žele to koristiti. Tradicionalno, TCP/IP mreže signaliziraju zagušenje tako što brišu pakete.
Kada je ECN uspješno dogovoren, ECN omogućeni ruter može postavljati bite u IP zaglavlju, umjesto odbacivanja paketa, da bi signalizirao početak zagušenja. Prijemnik paketa šalje odgovor sa indikacijom zagušenja predajniku, koji mora reagovati kao da je detektovao gubitak paketa. ECN koristi dva bita u polju rezervisanih bita u IP

zaglavlju. Ova dva bita se mogu koristiti za kodiranje jedne od sljedećih vrijednosti: ECN-unaware (ECN-onemogućen), ECN-aware (ECN-omogućen) ili Congestion Experienced (desilo se zagušenje).

Dodatno ECN bitima u IP zaglavlju, TCP koristi dvije zastavice (eng. flag) u TCP zaglavlju (rezervisani biti koji se nalaze poslije polja dužine zaglavlja) da bi signalizirao predajniku da smanji količinu podataka koju šalje. To su biti „ECN-echo“ i „Congestion Window Reduced“ (smanjen prozor zagušenja). Korištenje ECN-a kod TCP konekcije je opcionalno.

2.1.2. *Random Early Detection* (RED)

Nasumična rana detekcija (eng. Random Early Detection, RED) pokušava kontrolisati prosječnu veličinu redova čekanja tako što indicira krajnjim hostovima kada trebaju privremeno smanjiti brzinu transmisije paketa [4].

RED preuzima prednosti mehanizama kontrole zagušenja kod TCP-a. Time što slučajno briše pakete prije perioda velikog zagušenja, RED ukazuje izvorištu paketa da smanji brzinu transmisije. Ako se pretpostavi da izvorište paketa koristi TCP, on će smanjiti svoju brzinu transmisije sve dok svi paketi ne dostignu odredište, te tako indiciraju da je zagušenje otklonjeno. RED se može koristiti kao način da se prouzrokuje da TCP smanji brzinu transmisije paketa. Pri tome, TCP se nakratko pauzira, te nastavlja ubrzo zatim i prilagođava svoju brzinu transmisije tako da to mreža može da podrži.

RED distribuira gubitke u vremenu i održava male redove čekanja. Kada se desi zagušenje, RED počinje odbacivati pakete i to brzinom koja se izabere za vrijeme konfiguracije. RED odluči da odbaci paket kad je red čekanja određene veličine, te distribuira gubitke u vremenu (slučajnim odbacivanjem, u različitim vremenskim trenucima, paketa različitih konekcija), To odbacivanje paketa zapravo rade RED omogućeni ruteri.

2.1.3. BLUE, SFB i CHOKe

BLUE [5] koristi gubitak paketa i historiju korištenja linka kako bi upravljao redom čekanja. Radi tako što slučajno briše ili ECN-markira pakete u redu čekanja na ruteru prije nego što red postane prepun. BLUE red održava vjerovatnoću brisanja/markiranja p, te na osnovu ove vjerovatnoće briše ili markira pakete kada oni ulaze u red čekanja. Kada se red popuni, p se povećava za vrijednost konstante p_d, a kada se red isprazni, p se smanjuje za vrijednost konstante $p_i<p_d$. Ako red briše pakete zbog popunjenosti bafera, BLUE povećava vrijednost p, tako da povećava brzinu kojom šalje nazad notifikacije zagušenja ili kojom briše pakete.
Glavni nedostatak BLUE algoritma je što ne razlikuje protoke, već tretira sve protoke skupa kao jedan agregirani. Stoga, može se desiti da jedan agresivni protok izbaci iz reda pakete od drugog, manje agresivnog protoka.

SFB (Stochastic Fair BLUE) je stohastički pravedna varijanta BLUE algoritma, koja održava različitu vjerovatnoću brisanja/markiranja za svaki protok [6]. Time se rješava

glavni nedostatak kod BLUE algoritma, tako da SFB može pružiti pravednu raspodjelu prostora u baferu za svaki protok.

CHOKe (CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows) je algoritam koji radi tako što uspoređuje dolazni paket sa slučajno izabranim paketom u redu čekanja, te ukoliko pripadaju istom protoku, oba paketa se odbacuju [7]. Pretpostavka algoritma je da veliki protoci imaju i veliki broj paketa u redu čekanja. Međutim, kada se radi o velikom broju protoka u odnosu na ukupnu veličinu bafera, to predstavlja svega nekoliko paketa po protoku, te je time funkcionisanje algoritma limitirano.

2.2. Kontrola zagušenja na transportnom sloju

Metode za kontrolu zagušenja na transportnom sloju se provode na krajnjim uređajima u mreži, odnosno na samim hostovima koji su izvorišta ili odredišta komunikacija.

2.2.1. Tahoe TCP

TCP se zasniva na principu "očuvanja paketa" (eng. conservation of packets): ako konekcija radi na raspoloživom kapacitetu propusnog opsega (eng. bandwidth), onda se paket ne ubacuje u mrežu sve dok drugi paket ne napusti mrežu. TCP implementira navedeni princip tako što koristi potvrde prijema da tempira odlazeće pakete, zbog toga što potvrda znači da je određeni paket napustio mrežu. Isto tako, TCP održava prozor zagušenja (eng. congestion window, cwnd) da bi predstavio mrežni kapacitet [2]. Međutim, postoje određena pitanja koja trebaju biti riješena da bi se osigurala spomenuta ravnoteža:

- Odrediti raspoloživi propusni opseg;
- Osigurati da se ravnoteža održi;
- Odgovoriti na zagušenje.

Slow Start:
Transmisije paketa kod TCP-a su tempirane dolazećim potvrdama. Međutim, postoji problem kada se konekcija prvi put uspostavlja, jer da bi imali potvrde moraju se imati podaci u mreži, a da bi se ubacili podaci u mrežu moraju se imati potvrde. Da bi se prevazišao opisani problem, Tahoe TCP, u skladu sa predstavljenim u [2], nalaže da se prođe kroz proceduru zvanu *Slow Start*, bilo da se TCP konekcija tek starta, ili da se restarta poslije gubitka paketa. Razlog za ovu proceduru je što bi inicijalni nagli protok (eng. burst) mogao preplaviti mrežu i konekcija se možda nikada ne bi ni startala.
Slow Start nalaže da predajnik postavi prozor zagušenja na 1, a zatim za svaku primljenu potvrdu prijema od druge strane (eng. acknowledgement, ACK) povećava se cwnd za 1, tako da se u prvom RTT-u (eng. round-trip time - vrijeme povratnog puta) šalje 1 paket, u drugom se šalje 2, a u trećem se šalje 4.
Prozor zagušenja se povećava eksponencijalno sve dok se ne izgubi paket, što je znak da se desilo zagušenje. Kada se desi zagušenje, smanjuje se brzina slanja i reducira prozor zagušenja na 1. Nakon toga, ponovo se sve starta iz početka. Važna činjenica je što Tahoe TCP detektuje gubitke paketa zahvaljujući vremenskim ograničenjima (eng. timeout), koja ukazuju da je istekao vremenski interval unutar kojeg je trebala stići potvrda. Kod običnih implementacija, javljaju se vremenska ograničenja koja traju duže

vremena, tako da se može desiti da prođe prilično vremena prije nego što se primjeti gubitak paketa i retransmituje isti.

Congestion Avoidance:
Za *Congestion Avoidance* Tahoe TCP koristi "dodatna povećanja i višestruka smanjenja" (eng. Additive Increase Multiplicative Decrease). Gubitak paketa se uzima kao znak zagušenja i Tahoe TCP snima polovinu trenutnog prozora kao graničnu vrijednost (eng. threshold, ssthresh). Zatim, postavlja cwnd na 1 i starta *Slow Start* sve dok ne dostigne graničnu vrijednost. Poslije toga, inkrementira se linearno sve dok se ne desi gubitak paketa. Tako se prozor povećava polako dok se približava kapacitetu propusnog opsega.

Problemi:
Tahoe TCP ima problem zbog toga što je potreban kompletan interval vremenskog ograničenja da bi se detektovao gubitak paketa, s tim da kod većine implementacija treba duže vremena zbog dugotrajnih vremenskih ograničenja. Isto tako, ne šalju se ACK-ovi istog trenutka kad se dobije paket, već se šalju kumulativne potvrde. Tako da svaki put kada se paket izgubi čeka se da istekne vremensko ograničenje i protočna struktura (eng. pipeline) se isprazni. To predstavlja najveću cijenu kada je u pitanju kašnjenje na linkovima sa velikim propusnim opsegom.

2.2.2. Reno TCP

Reno TCP, dokumentovan u [8], zadržava osnovne principe Tahoe algoritma, kao što su *Slow Start* i dugotrajni retransmisijski tajmer (eng. timer). Međutim, Reno TCP dodaje više inteligencije tako što ranije detektuje izgubljene pakete i protočna struktura se ne prazni svaki put kada se izgubi paket. Reno TCP zahtijeva da se odmah dobije potvrda svaki put kada segment stigne na odredište. Logika iza ovoga je da svaki put kada stigne dupla potvrda, ista može biti primljena ako je sljedeći segment koji se očekuje u sekvenci zakašnjen u mreži, a segmenti koji su stigli su izvan redoslijeda, ili pak ako je paket izgubljen. Ako se primi dovoljan broj duplih potvrda, onda to znači da je prošlo dovoljno vremena i da čak ako je segment krenuo dužim putem, trebao je već stići do prijemnika. Postoji velika vjerovatnoća da je paket izgubljen. Zbog toga, Reno TCP sugeriše algoritam koji sa naziva *Fast Retransmit* [9]. Svaki put kada se prime 3 duplicirana ACK-a, to se uzima kao znak da je segment izgubljen, tako da se retransmituje segment bez čekanja da istekne vremensko ograničenje. Tako se uspjeva retransmitovati segment sa protočnom strukturom koja je skoro puna.
Druga modifikacija koju Reno TCP čini je da poslije gubitka paketa ne smanjuje prozor zagušenja na 1. Razlog je što bi to ispraznilo protočnu strukturu. Reno TCP ulazi u algoritam koji se naziva *Fast Retransmit*. Osnovni algoritam je kao što slijedi:
Svaki put kada se prime 3 duplicirana ACK-a, uzima se da to znači da je segment izgubljen, te se retransmituje segment istog trena i ulazi u *Fast Recovery*. Postavlja se ssthresh na polovinu trenutne veličine prozora i cwnd na istu vrijednost. Za svaki primljeni duplicirani ACK povećava se cwnd za 1. Ako je povećani cwnd veći nego količina podataka u protočnoj strukturi, onda se transmituje novi segment, inače se čeka. Ako ima "w" segmenata unutar prozora i jedan je izgubljen, primit će se (w-1) dupliciranih ACK-ova. Pošto je cwnd smanjen na w/2, polovina prozora sa podacima je potvrđena prije nego što se može slati novi segment. Kada se transmituje segment, mora se čekati najmanje jedan RTT prije nego što se primi nova potvrda. Svaki put kada se primi novi ACK smanjuje se cwnd na ssthresh. Ako je prije toga stiglo (w-1)

dupliciranih ACK-ova, onda u tom trenutku bi trebalo biti tačno w/2 segmenata u protočnoj strukturi, što je jednako postavljenom cwnd-u na kraju *Fast Recovery*-a. Tako se ne prazni protočna struktura, već se samo smanjuje protok. Nakon toga se nastavlja sa *Congestion Avoidance*-om kao kod Tahoe algoritma.

Problemi:
Reno TCP se ponaša veoma dobro kada se radi o malim gubicima paketa. Ali, kada postoje višestruki gubici paketa u istom prozoru, onda se Reno TCP ne ponaša previše dobro i njegove performanse su približno iste kao kod Tahoe TCP-a u istim uslovima. Razlog za to je što može otkriti samo pojedinačne gubitke paketa. Ako postoje višestruki gubici paketa, onda prva informacija o gubitku paketa dolazi kada se prime dupli cirani ACK-ovi. Ali, informacija o drugom izgubljenom paketu će doći tek nakon što ACK za prvi retransmitovani segment dostigne predajnik poslije jednog RTT-a.
Takođe, moguće je da cwnd bude smanjen dva puta zbog gubitaka paketa koji su se desili u istom prozoru. Drugi problem je što ako je prozor veoma mali kada se desi gubitak, onda se nikada neće dobiti dovoljno duplih potvrda za *Fast Retransmit* i morati će se čekati da istekne dugotrajno vremensko ograničenje.

2.2.3. New Reno TCP

New Reno TCP, opisan u [10], predstavlja manju modifikaciju urađenu u odnosu na Reno TCP. U stanju je da otkrije višestruke gubitke paketa, tako da je mnogo efikasniji nego Reno TCP u slučaju višestrukih gubitaka paketa.
Kao i Reno, New Reno TCP takođe ulazi u *Fast Retransmit* kada primi višestruke dupli cirane pakete, ali se razlikuje od Reno TCP-a po tome što ne izlazi iz *Fast Recovery*-a sve dok svi podaci koji su bili poslati u vrijeme kada je ušao u *Fast Recovery* ne budu potvrđeni. Tako izbjegava problem koji ima Reno TCP sa smanjivanjem cwnd prozora više puta.
Fast Retransmit faza je ista kao kod Reno TCP-a. Razlika je u *Fast Recovery* fazi koja dozvoljava višestruke retransmisije kod New Reno TCP-a. Svaki put kada New Reno uđe u *Fast Recovery*, pri tome bilježi maksimalni preostali segment. *Fast Recovery* faza se dešava kao kod Reno TCP-a, ali kada se primi novi ACK onda postoje dva slučaja:
Ako ACK potvrđuje sve segmente koji su bili preostali kada je New Reno ušao u *Fast Recovery,* onda New Reno izlazi iz *Fast Recovery*-a i postavlja cwnd na ssthresh, te nastavlja *Congestion Avoidance* kao Tahoe TCP;
Ako ACK predstavlja samo parcijalni ACK, onda New Reno TCP zaključuje da je sljedeći segment u nizu izgubljen i retransmituje segment, te postavlja broj primljenih dupliciranih ACK-ova na nula.
New Reno TCP izlazi iz *Fast Recovery*-a kada su svi podaci u prozoru potvrđeni.

Problemi:
New Reno ima problem što mu treba jedan RTT da detektuje svaki gubitak paketa. Kada je ACK za prvi retransmitovani segment primljen, tek tada se može zaključiti koji je drugi segment izgubljen.

2.2.4. SACK TCP

TCP sa "selektivnim potvrdama" (eng. Selective ACKnowledgments) je proširenje u odnosu na Reno TCP i radi na problemima sa kojima se susreću Reno TCP i New Reno TCP, a posebno na detekciji višestrukih izgubljenih paketa, kao i retransmisiji više od jednog izgubljenog paketa po RTT-u.
SACK TCP zadržava *Slow Start* i *Fast Retransmit* dijelove od Reno TCP-a [11]. Isto tako, ima dugotrajna vremenska ograničenja kao i Tahoe TCP, u slučaju da gubitak paketa ne bude detektovan od strane modifikovanog algoritma.
SACK TCP zahtijeva da se segmenti ne potvrđuju kumulativno, već da budu potvrđeni selektivno. Tako svaki ACK ima blok koji opisuje koji segmenti se potvrđuju. Na osnovu toga predajnik ima sliku o tome koji segmenti su potvrđeni, a koji su još preostali. Svaki put kada predajnik uđe u *Fast Recovery*, pri tome inicijalizira varijabilnu protočnu strukturu (eng. variable pipe) koja procijenjuje koliko podataka je preostalo u mreži, te postavlja cwnd na polovinu svoje vrijednosti. Svaki put kada primi ACK smanjuje protočnu strukturu za 1, a svaki put kada retransmituje segment povećava istu za 1. Svaki put kada protočna struktura postane manja od cwnd prozora, SACK TCP provjerava koji segmenti nisu primljeni i ponovo ih otprema. Ako nema takvih preostalih segmenata, onda šalje novi paket. Tako se postiže da više od jednog izgubljenog segmenta može biti poslato u jednom RTT-u.

Problemi:
Najveći problem kod SACK TCP-a je što trenutno selektivne potvrde nisu podržane od strane prijemnika. Implementirati SACK TCP, odnosno implementirati selektivne potvrde, nije lak zadatak.

2.2.5. Vegas TCP

Vegas TCP je implementacija TCP-a koja predstavlja modifikaciju od Reno TCP [12]. Izgrađena je na činjenici da su proaktivne mjere da se spriječi zagušenje mnogo efikasnije nego reaktivne. Vegas TCP pokušava iznaći rješenje problema dugotrajnih vremenskih ograničenja tako što sugeriše algoritam koji provjerava vremenska ograničenja veoma efikasnim rasporedom. Isto tako, prevazilazi problem sa zahtijevanjem dovoljnog broja duplih potvrda da bi se detektovao gubitak paketa, a sugeriše i modifikovani *Slow Start* algoritam koji sprječava od zagušenja mreže. Pri tome, ne zavisi samo od gubitka paketa kao znaka da se desilo zagušenje. Naprotiv, detektuje zagušenje prije nego što se desi gubitak paketa. Pored toga, zadržava i dalje mehanizme od Tahoe TCP-a i Reno TCP-a, tako da gubitak paketa i dalje može biti detektovan zahvaljujući dugotrajnim vremenskim ograničenjima.

Tri glavne promjene koje uvodi Vegas TCP su sljedeće:
Novi retransmisijski mehanizam:
Vegas TCP proširuje retransmisijski mehanizam od Reno TCP-a. Bilježi kada je svaki segment poslat i računa procjenu RTT-a tako što bilježi koliko treba vremena potvrdi da se vrati. Svaki put kada je primljena dupla potvrda, provjerava se da li je (trenutno vrijeme – transmisijsko vrijeme segmenta) > procijenjenog RTT-a; ako je to slučaj onda se istog trena retransmituje segment bez čekanja da stignu tri duple potvrde ili da istekne vremensko ograničenje (eng. timeout). Tako prevazilazi problem sa kojim se susreće Reno TCP kada nije u stanju da detektuje gubitak paketa kada ima mali prozor i

ne primi dovoljan broj dupliciranih ACK-ova. Da bi zahvatio ostale segmente koji su možda izgubljeni prije retransmisije, kada stigne neduplicirana potvrda, ako je to prva ili druga poslije nove potvrde, onda se ponovo provjerava vrijednost timeout-a. Ako vrijeme od kako je segment poslat prevazilazi vrijednost timeout-a, onda se retransmituje segment bez čekanja za duplom potvrdom. Tako Vegas TCP može detektovati višestruke gubitke paketa.
Isto tako, Vegas TCP smanjuje svoj prozor samo ako je retransmitovani segment poslat poslije zadnjeg smanjenja. Time, Vegas TCP prevazilazi problem koji je imao Reno TCP sa smanjivanjem prozora zagušenja više puta kada se izgubi više paketa.

Congestion Avoidance:
Vegas TCP se razlikuje od svih ostalih implementacija po svom ponašanju za vrijeme faze *Congestion Avoidance*. Ne koristi gubitak segmenta da bi signalizirao da postoji zagušenje. Određuje zagušenje na osnovu smanjenja brzine slanja u odnosu na očekivanu brzinu, a sve to kao rezultat velikih redova čekanja koji se stvaraju na ruterima. Tako svaki put kada je izračunata brzina previše daleko od očekivane, povećava transmisiju tako da iskoristi raspoloživi propusni opseg (eng. bandwidth), a svaki put kada izračunata brzina dođe previše blizu očekivane vrijednosti smanjuje se transmisija da bi se spriječila saturacija (zasićenost) bandwidtha. Tako se Vegas TCP suočava sa zagušenjem prilično efikasno, jer ne troši nepotrebno bandwidth tako što bi transmitovao na previše velikoj brzini i stvarao zagušenje, a zatim to prekidao i vraćao se nazad, što drugi algoritmi rade.

Modifikovani *Slow Start*:
Vegas TCP se razlikuje od ostalih algoritama za vrijeme svoje *Slow Start* faze. Razlog za ovu modifikaciju je što kada se konekcija po prvi put starta Vegas TCP ne zna koliko ima raspoloživog bandwidtha, tako da je moguće da za vrijeme faze eksponencijalnog rasta previše optereti bandwidth velikom količinom podataka i time prouzrokuje zagušenje. Zbog toga, Vegas TCP povećava eksponencijalno svakog RTT-a, a između toga računa aktuelnu propusnost i poredi je sa očekivanom, te kada razlika postane veća od određenog *threshold*-a izlazi iz *Slow Start*-a i ulazi u fazu *Congestion Avoidance*.

2.2.6. Fast TCP

Fast TCP primjenjuje algoritam kontrole zagušenja koji je specijalno namijenjen za linkove velikih brzina i velikih udaljenosti [13]. Kompatibilan je sa postojećim TCP algoritmima, tako da zahtjeva modifikaciju samo na strani predajnika.

Fast TCP nastoji da održi konstantnim broj paketa u redovima čekanja kroz cijelu mrežu. Broj paketa u redovima se procijenjuje tako što se mjeri razlika između izmjerenog RTT-a i baznog RTT-a. Bazni RTT je procijenjen kao minimalni ustanovljeni RTT za datu konekciju, tako što je izmjeren kada nije bilo redova čekanja.

Ako je premalo paketa u redu, onda se brzina slanja povećava. Ako je previše paketa u redu, onda se brzina smanjuje. U tom pogledu je Fast TCP veoma sličan Vegas TCP-u. Razlika između Vegas TCP-a i Fast TCP-a leži u tome kako se brzina prilagođava kada je broj pohranjenih paketa previše mali ili previše veliki. Vegas TCP podešava brzinu

fiksnim koracima, nezavisno od toga koliko je daleko trenutna brzina od ciljane brzine. Fast TCP pravi veće korake kada je sistem daleko od ravnoteže, a manje kada je blizu. To poboljšava brzinu konvergencije i stabilnost.

2.2.7. HS TCP

HighSpeed TCP je verzija TCP-a koji reaguje bolje kada se koriste veliki prozori zagušenja na mrežama sa velikim *bandwidth*-om. Dokumentovan je u [14]. Razlog zbog kojeg performanse TCP-a počinju da opadaju iznad 100 Mb/s ima veze sa algoritmom podešavanja prozora. U svojoj *Congestion Avoidance* fazi, obični TCP povećava svoj prozor na strani predajnika za jedan paket svakog RTT-a. Kada detektuje zagušenje, onda smanjuje prozor na polovinu. Za konekcije velike brzine, optimalna veličina prozora treba biti minimalno 8000 paketa. To znači da treba 4000 RTT-ova da se oporavi od zagušenja. Ako svaki RTT traje 100 ms, onda je vrijeme oporavka 400 s.

HS TCP mijenja način na koji je prozor otvoren svakog RTT-a i zatvoren u slučaju zagušenja, a sve to u funkciji apsolutne veličine prozora. Kada je prozor mali, HS TCP se ponaša tačno kao obični TCP. Ali, kada je prozor veliki, povećava se za veću vrijednost, a smanjuje se za manju vrijednost, gdje su te veličine izabrane na osnovu precizne vrijednosti prozora u radu. Efekat opisanih promjena se ogleda u tome da sporost običnog TCP-a nestaje.

Manje smanjenje i veće povećanje veličine prozora znači da se HS TCP oporavlja brže nego obični TCP. Ovakav pristup omogućava potpuno iskorištenje WAN linkova velikih brzina. Sve radi ispravno čak i kada HS TCP konekcije dijele WAN linkove sa običnim TCP konekcijama.

2.2.8. Africa TCP

Africa TCP (Adaptive and Fair Rapid Increase Congestion Avoidance mechanism for TCP – adaptivan i pravedan mehanizam izbjegavanja zagušenja sa brzim povečanjem) je opisan u [15]. Africa TCP je hibridni protokol koji koristi metriku kašnjenja da bi odredio da li je link zagušen. U slučaju da nema zagušenja, fast mode koristi agresivno, skalabilno pravilo izbjegavanja zagušenja. Kada nastupi zagušenje, slow mode prebacuje na konzervativnije Reno TCP pravilo izbjegavanja zagušenja.

Pod povoljnim uslovima kašnjenja, koji odgovaraju prisustvu slobodnog raspoloživog bandwidth-a, protokol povečava svoj prozor na agresivan, skalabilan način. Kada je u „fast“ modu, koriste se HS TCP parametri povečanja i smanjenja.

2.2.9. Yeah TCP

Yeah TCP (Yet Another TCP Protocol) ima dva različita načina rada: „brzi“ (eng. fast) i „spori“ (eng. slow) režim, kao i kod Africa TCP-a. Predstavljen je u [16]. Za vrijeme „brzog“ režima, Yeah TCP inkrementira svoj prozor zagušenja u skladu sa agresivnim algoritmom (npr. STCP). U „sporom“ režimu, Yeah TCP se ponaša kao Reno TCP.

Stanje u kojem će se nalaziti Yeah TCP zavisi od procijenjenog broja paketa u redu čekanja koji predstavlja usko grlo (eng. bottleneck queue). Za vrijeme „Slow“ moda, implementiran je preventivni algoritam za smanjenje zagušenja koji radi tako što smanjuje prozor zagušenja. Preventivno smanjenje zagušenja sprječava da red čekanja, koji predstavlja usko grlo, ne bi previše narastao, smanjujući pri tome kašnjenja kod redova čekanja i gubitke paketa zbog prepunog bafera (eng. buffer overflow). Kao što je prikazano u [12], preventivno smanjenje zagušenja je optimalno samo kada se protoci, koji to implementiraju, ne natječu sa „pohlepnim“ (eng. greedy) izvorima, kao što su to stare verzije TCP protokola (eng. legacy TCP). Preventivno smanjenje zagušenja nije u stanju da se natječe sa „pohlepnim“ ili „grabljivim“ protocima jer ima tendenciju da prepusti bandwidth takvim „grabljivim“ izvorima i da u potpunosti ostane bez bandwidth-a. Da bi izbjeglo nepravedno takmičenje sa „legacy“ protocima, Yeah TCP implementira mehanizam kako bi detektovao da li se takmiči sa „pohlepnim“ izvorima.

2.2.10. Pravednost i *less-than-best-effort* servis

U radu [17] autori predlažu pristup koji podrazumijeva algoritam kontrole zagušenja baziran na kašnjenju, a koji ima ugrađenu pravednost spram TCP tokova baziranih na gubicima. To se postiže tako što se predloženi algoritam ponaša kao da je baziran na gubicima kada se primjeti da postoje TCP tokovi bazirani na gubicima.

U radu [18] iznesen je pregled pristupa vezano za TCP protokol koji podrazumijevaju „*less-than-best-effort*“ ili „*lower than best-effort***“** servis. To se odnosi na način rada TCP protokola tako da ima manji uticaj na bandwidth nego standardni TCP, odnosno da ima što manji uticaj na druge konkurentne tokove. Ovakvi pristupi se mogu dobro pokazati za aplikacije koje kreiraju saobraćaj koji nije previše urgentan, kao što su npr. automatski backup, softverski update-i, ili peer-to-peer aplikacije. Primjer verzije TCP protokola koja je u skladu sa prethodno opisanim principom je već predstavljeni Vegas TCP.

Pitanje pravednosti i koegzistencije različitih TCP tokova je veoma bitno što pokazuju i radovi na datu temu [19], [20], [21] i [22].

2.2.11. Multi-path TCP

Korištenje više paralelnih TCP tokova je predloženo u [23]. To se odnosi na razdvajanje jednog TCP toka na više TCP tokova, pri čemu oni trebaju ići različitim putevima. Ideja je da zagušenje može nastati na jednom putu, ali da će na drugom putu vjerovatno biti bolja situacija.

Kritika gore spomenutog Multi-path TCP rješenja se iznosi u [24], jer na osnovu prethodnog istraživanja nije dokazan benefit uvođenja MP TCP-a. Autori rada nude i vlastito rješenje u vidu poboljšanog algoritma za koji tvrde da je optimalan.

Problematikom paralelnih, odnosno višestrukih TCP tokova se bave mnogi radovi [25]-[31], u kojima se iznose prednosti datog pristupa i moguća poboljšanja.

2.2.12. TCP za real time saobraćaj

TCP se dugo vremena smatrao neprikladnim za real-time aplikacije zbog svojih mehanizama potvrde prijema i retransmisije izgubljenih segmenata, što utiče na brzinu rada. Međutim, zbog svojih brojnih prednosti, kao i zbog činjenice da je UDP često blokiran na firewall-ima na ulazu u korporativne mreže, sve je veća njegova primjena i za voice (npr. Skype) i video (npr. Youtube) saobraćaj.

U radu [32] se iznose rezultati istraživanja koje pokazuje da jednostavne šeme na aplikativnom nivou, kao što su dijeljenje paketa i paralelne konekcije, mogu smanjiti kašnjenje real-time TCP tokova za 30%, odnosno 90%.

U radovima [33]-[36] se iznose analize i razmatranja vezano za video streaming preko HTTP protokola, odnosno upotrebom TCP protokola.

3. Osnovni ciljevi i plan istraživanja

Ciljevi istraživanja su da se pokažu mogući mehanizmi za poboljšanje kontrole zagušenja na Internetu, kao i da se pokušaju iznaći poboljšanja postojećih mehanizama kontrole zagušenja. Prvenstveno, cilj je da se iznađu moguća poboljšanja kontrole zagušenja kod TCP protokola, na osnovu matematičke optimizacije parametara konekcije, a u skladu sa principom heurističkog algoritma. Istraživanje će se fokusirati na pronalaženje metode koja omogućava da se parametri, prilikom startanja TCP konekcije, podešavaju na osnovu heurističke strategije. Tražile bi se optimalne vrijednosti koje su relativne u odnosu na vrijednosti prethodnih konekcija.

U samom pristupu za rješavanje gore navedenog problema bi se razmotrila i implementacija ideje dualizma, odnosno mogućnosti koegzistencije različitih TCP verzija, za različite namjene, od kojih jedna može biti HD video. Istražila bi se mogućnost realizacije pristupa koji bi se ogledao u tome što bi sadržavao dva različita TCP algoritma: npr. jedan za HD video emitovanje, a drugi za potrebe ostalih konekcija. Kada bi se trebao prenositi HD video, pokretao bi se modifikovani (agresivniji) TCP. U ostalim slučajevima bi se koristio TCP algoritam sa ugrađenim principom pravednosti spram drugih tokova (eng. „fairness“).

Plan istraživanja:

1) Pročitati relevantne naučne radove iz date oblasti, sa posebnim osvrtom na najnovije radove (2013.)
2) Formirati idejni koncept (2013.)
3) Uspostaviti testnu platformu (2013/2014)
4) Pripremiti i odbraniti projekat (maj/juni 2014.)
5) Izraditi predloženi model/algoritam (2014.)
6) Testirati predloženi model/algoritam (2014.)
7) Objaviti rezultate istraživanja na konferencijama sa IEEE referencom (2014/2015)
8) Napisati i odbraniti doktorski rad (2014/2015)

4. Metodologija istraživanja

Metodologija istraživanja će se sastojati od analize problema, modeliranja i simulacije u cilju prepoznavanja problema i validacije rješenja.
Analiza problema će se izvesti u cilju iznalaženja rješenja u skladu sa ciljevima istraživanja. U tu svrhu će se izvršiti analiza problematike zagušenja u mreži baziranoj na IP protokolu, a zatim analiza mehanizama kontrole mrežnog zagušenja.

Nakon analize izvršit će se modeliranje mehanizma za kontrolu zagušenja, gdje će se i definisati ključni parametri modela. Zadatak je pronaći odnose između ključnih parametara modela kako bi imali što manje zagušenja u mreži. Tokom faze modeliranja koristiti će se različiti modeli kontrole zagušenja u cilju rješavanja postavljenog problema.

Simulacija kao eksperimentalni dio istraživanja će se provesti sa ciljem prepoznavanja i demonstracije problema, kao i validacije modeliranog rješenja. Simulacije će se izvoditi na nekom od dostupnih simulatora računarskih mreža i komunikacijskih protokola, kao što je npr. OPNET. Pri tome se mogu vršiti pojedinačne simulacije da bi se pokazala poboljšanja na osnovu pojedinačnih mehanizama. Isto tako, mogu se vršiti i komparativne simulacije koje međusobno porede više različitih varijanti, da bi se iznašla najbolja varijanta sa aspekta mrežnih performansi.

5. Očekivani izvorni naučni doprinos disertacije

a) Prijedlog metode mogućeg poboljšanja mehanizama kontrole zagušenja kod TCP protokola, bazirane na matematičkoj optimizaciji parametara konekcije, a u skladu sa principom heurističkog algoritma.

b) Dizajn simulacije koji će prikazati različite implementacije mehanizama kontrole zagušenja, a naročito metode predložene za poboljšanje kontrole zagušenja, te omogućiti komparativnu analizu prednosti i nedostataka pojedinačnih rješenja.

c) Analiza zbirnih rezultata i moguća poboljšanja sveukupnih mehanizama kontrole zagušenja.

6. Sadržaj rada:

1 UVOD

2 INTERNET I KONTROLA ZAGUŠENJA

3 KONTROLA ZAGUŠENJA NA TRANSPORTNOM SLOJU

4 KONTROLA ZAGUŠENJA NA INTERNET/NETWORK SLOJU

5 MODEL PREDLOŽENOG RJEŠENJA

6 VERIFIKACIJA PREDLOŽENOG RJEŠENJA

7 ZAKLJUČAK

8 LITERATURA

7. Prijedlog okvirnog sadržaja rada po poglavljima

1. Uvod

U uvodnom dijelu će biti iznesen pregled zahtjeva koji se postavljaju pred Internet i TCP/IP protokol. Bit će opisani problemi i ograničenosti s kojima se TCP/IP protokol može susretati, s obzirom na današnje okruženje u kojem djeluje. Bit će pomenuta potreba za mehanizmima kontrole zagušenja i dosadašnja rješenja koja su podržala ovu problematiku.

2. Internet i kontrola zagušenja

U ovom poglavlju će biti opisana arhitektura Interneta. Bit će prikazan TCP/IP model protokola, kao i njegovo poređenje sa referentnim OSI modelom. U sklopu predstavljanja TCP/IP modela, bit će opisani i mrežni protokoli koji rade na njegovim pojedinačnim slojevima.

U nastavku poglavlja će biti predstavljen problem zagušenja na Internet mreži, kao i uopšteni načini na koje se sprovodi kontrola zagušenja. To se prvenstveno odnosi na opšte mehanizme kontrole zagušenja, a koji se koriste od strane specifičnih algoritama, kao što su npr. različite varijante TCP protokola.

3. Kontrola zagušenja na transportnom sloju

U ovom poglavlju će biti detaljno opisani mehanizmi kontrole zagušenja koji se koriste na transportnom sloju. Bit će predstavljene različite varijante TCP protokola, te opisane njihove prednosti, kao i eventualni nedostaci.

U prvom dijelu poglavlja će biti opisane varijante TCP protokola bazirane na gubicima paketa/segmenata kao metodi detekcije zagušenja, kao što su npr. Tahoe TCP, Reno TCP, New Reno TCP, SACK TCP, HS TCP, itd.

Zatim će biti opisane varijante TCP protokola bazirane na kašnjenju paketa/segmenata kao detekciji zagušenja, kao npr. Vegas TCP ili Fast TCP.

U zadnjem dijelu ovoga poglavlja će biti opisani kombinovani pristupi, odnosno varijante TCP protokola koje koriste i gubitke i kašnjenje kao indikaciju zagušenja, kao npr. Africa TCP ili Yeah TCP.

4. Kontrola zagušenja na Internet/network sloju

U ovom dijelu će biti predstavljene mogućnosti kontrole zagušenja na nivou IP protokola. Bit će detaljno opisana tzv. Eksplicitna notifikacija o zagušenju (ECN: Explicit Congestion Notification). Kada se uoči zagušenje negdje na putu kroz mrežu, postavlja se specijalno polje u zaglavlju IP paketa kako bi se ukazalo krajnjim sistemima da treba prilagoditi transmisiju.

Dodatno tome, na IP nivou, može se vršiti odbacivanje paketa kako bi se izbjeglo zagušenje. Radi se o tzv. Algoritmima za aktivno upravljanje redovima čekanja (eng. Active Queue Management, AQM). Najpoznatiji od njih je Nasumično rano odbacivanje (eng. Random Early Drop, RED). Postoje i drugi algoritmi, kao npr. Drop Tail, BLUE, SFB (Stochastic Fair BLUE), itd.

5. Model predloženog rješenja

U ovom dijelu će biti predložen novi pristup koji bi se mogao koristiti sa ciljem poboljšanja kontrole zagušenja na nivou TCP protokola. Predloženi pristup će demonstrirati optimizaciju parametara TCP konekcije pomoću matematičkih metoda iz oblasti Operacionih istraživanja ili Vještačke inteligencije. Radi se o metodi mogućeg poboljšanja mehanizama kontrole zagušenja kod TCP protokola, bazirane na matematičkoj optimizaciji parametara konekcije, a u skladu sa principom heurističkog algoritma. Pri tome se vrijednosti parametara modifikuju prilikom startanja svake naredne konekcije, relativno u odnosu na prethodne vrijednosti, a tražeći optimalno rješenje na osnovu heurističke strategije.

Predložena metoda sadrži mogućnost učenja: algoritam treba odrediti sljedeću vrijednost parametara u smislu pronalaska optimalnog rješenja. Pri tome se algoritam može smatrati heurističkim jer prolazi fazu učenja, tj. samoprilagođavanja, nakon čega prelazi u operativni rad. Isto tako, algoritam može sačuvati izračunate vrijednosti parametara za sljedeću konekciju koja se bude startala. Pošto se cijeli proces ponavlja pri svakom startanju nove konekcije, to se može smatrati dugoročnim učenjem. Takav pristup omogućava algoritmu da nauči kakvo je stanje mreže i da takvo znanje sačuva za buduću upotrebu.

Bit će razmotrena implementacija ideje dualizma, odnosno mogućnost koegzistencije različitih TCP verzija, za različite namjene, od kojih jedna može biti HD video. Pri tome će se istražiti mogućnost realizacije pristupa koji sadrži dva različita TCP algoritma: npr. jedan za HD video emitovanje, a drugi za potrebe ostalih konekcija. Kada se treba prenositi HD video, pokreće se modifikovani (agresivniji) TCP. U ostalim slučajevima se koristi TCP algoritam sa ugrađenim principom pravednosti spram drugih tokova (eng. „fairness“).

U svrhu evaluacije predloženog rješenja, bit će napravljena testna platforma koja povezuje OPNET simulator mrežnih protokola sa MATLAB razvojnim okruženjem. Na OPNET-u će se pokretati simulacije mrežnog saobraćaja, sa unaprijed određenim procentom gubitaka paketa, dok će se u MATLAB-u izvršavati skripta sa matematičkim algoritmom, koja će optimizirati parametre u samom OPNET simulatoru.

6. Verifikacija predloženog rješenja

Verifikacija predloženih rješenja će biti urađena pomoću simulacija. Simulacije će se izvoditi na nekom od dostupnih simulatora mrežnih protokola, kao što je npr. OPNET. Pri tome će se vršiti simulacije koje imaju za cilj pokazati poboljšanja na osnovu pojedinačnih mehanizama. Isto tako, vršiti će se i komparacije koje međusobno porede više različitih varijanti, a sve s ciljem da se iznađe najbolja varijanta sa aspekta mrežnih performansi.

Eventualno, provjera validnosti predloženog rješenja bi se mogla uraditi implementacijom na stvarnom sistemu (npr. na računarima sa Linux operativnim sistemom). Pri tome, provjera efikasnosti takvog sistema bi mogla biti na globalnom Internetu sa slučajnim gubicima, ili u lokalnoj računarskoj mreži sa kontrolisanim gubicima.

Naročitu primjenu predloženo rješenje bi moglo imati za HD video emitovanje putem HTTP/TCP/IP protokola. Zbog toga bi, eventualno, validacija mogla biti urađena i na stvarnom sistemu za video emitovanje putem Interneta.

7. Zaključno razmatranje

U zaključnom razmatranju će biti sumirani rezultati rada. Bit će spomenuti problemi sa zagušenjem koji se dešavaju na Internetu, kao i najčešće korištena rješenja. Takođe će biti navedene i beneficije koje donose mehanizmi kontrole zagušenja, te razmotrene potencijalne smjernice daljeg razvoja. Naročita pažnja će biti posvećena predloženom rješenju i napretku koje ono predstavlja u odnosu na postojeća rješenja.

8. Polazna literatura

1. **John Nagle,** "Congestion Control in IP/TCP Internetworks", RFC 896, January 1984
2. **Jacobson, V.,** "Congestion Avoidance and Control"**,** SIGCOMM Symposium on Cummunication Architecture and Protocols, 1988
3. **K. Ramakrishnan, S. Floyd, D. Black,** "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001
4. **Floyd, S., and Jacobson, V.,** "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993
5. **Wu-chang Feng, Kang G. Shin, Dilip D. Kandlur, Debanjan Saha**, "The BLUE Active Queue Management Algorithms", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 10, NO. 4, AUGUST 2002
6. **Wu-chang Feng, Dilip D. Kandlur, Debanjan Saha, Kang G. Shin**, "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness", IEEE INFOCOM 2001
7. **Rong Pan, Balaji Prabhakar, Konstantinos Psounis.** "CHOKE, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation". IEEE INFOCOM 2000
8. **W. Stevens,** "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms"**,** RFC 2001, January 1997
9. **V.Jacobson**, "Modified TCP Congestion Control and Avoidance Alogrithms". Technical Report 30,Apr 1990.
10. **S.Floyd, T.Henderson**, "The New-Reno Modification to TCP's Fast Recovery Algorithm", RFC 2582, Apr 1999.
11. **K.Fall, S.Floyd**, "Simulation Based Comparison of Tahoe, Reno and SACK TCP", ACM SIGCOMM Computer Communication Review Homepage archive, Volume 26, Issue 3, July 1996
12. **L.S.Brakmo, L.L. Peterson,** "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communication, vol. 13[1995],(1465-1490)
13. **Wei, D., Jin, C., Low, S., Hegde, S.,** "Fast TCP: Motivation, Architecture, Algorithms, Performance", IEEE INFOCOM 2004
14. **S. Floyd,** "Highspeed TCP for large congestion windows", RFC 3649, 2003 [Online: http://www.ietf.org/rfc/rfc3649.txt]
15. **Ryan King, Richard Baraniuk, Rudolf Riedi,** "TCP-Africa: An Adaptive and Fair Rapid Increase Rule for Scalable TCP", Departments of Electrical and Computer Engineering and of Statistics, Rice University, Houston Texas, IEEE INFOCOM 2005
16. **Andrea Baiocchi, Angelo P. Castellani, Francesco Vacirca,** "YeAH TCP: Yet Another Highspeed TCP", INFOCOM Department, University of Roma "Sapienza", Roma, Italy, Fifth International Workshop on Protocols for Fast Long-Distance Networks PFLDnet 2007
17. **Łukasz Budzisz, Rade Stanojevic, Arieh Schlote, Fred Baker, Robert Shorten,** "On the Fair Coexistence of Loss- and Delay-Based TCP", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 19, NO. 6, DECEMBER 2011

18. **David Ros, Michael Welzl,** "Less-than-Best-Effort Service: A Survey of End-to-End Approaches", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 2012
19. **A. Tang, J. Wang, S. Low, M. Chiang,** "Equilibrium of heterogeneous congestion control protocols", *Proc. 24th IEEE INFOCOM*, vol. 2, Mar. 2005,
20. **A. Tang, J. Wang, S. Low, M. Chiang,** "Equilibrium of heterogeneous congestion control: Existence and uniqueness", IEEE/ACM Trans. Netw., vol. 15, no. 4, pp. 824–837, Aug. 2007.
21. **A. Tang, X. Wei, S. Low, M. Chiang,** "Equilibrium of heterogeneous congestion control: Optimality and stability", IEEE/ACM Trans. Netw., vol. 18, no. 3, pp. 844–857, Jun. 2010.
22. **D. A. Hayes, G. Armitage,** "Improved coexistence and loss tolerance for delay based TCP congestion control", Proc. 35th Annu. IEEE LCN, pp. 24–31, Oct. 2010.
23. **Huaizhong Han, Srinivas Shakkottai, C. V. Hollot, R. Srikant, Don Towsley,** "Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the internet", IEEE/ACM Transactions on Networking (TON) , Volume 14 Issue 6, December 2006
24. **Ramin Khalili, Nicolas Gast,** "MPTCP is not Pareto-Optimal: Performance Issue and a Possible Solution", IEEE/ACM Transactions on Networking (TON) archive, Volume 21, Issue 5, October 2013
25. **D. Damjanovic, M. Welzl,** "MulTFRC: Providing weighted fairness for multimedia applications (and others too!)", *ACM SIGCOMM Computer Communications Review*, vol. 39, no. 9, Jul. 2009.
26. **D. Damjanovic, M. Welzl,** "An extension of the TCP steady-state throughput equation for parallel flows and its application in MulTFRC", *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1676–1689, Dec. 2011.
27. **S. Floyd, M. Handley, J. Padhye, J. Widmer,** "TCP Friendly Rate Control (TFRC): protocol specification", Internet Standards Track RFC 5348, IETF, Sep. 2008. [Online]. Available: http://tools.ietf.org/html/rfc
28. **J. Crowcroft, P. Oechslin,** "Differentiated end-to-end internet services using a weighted proportional fair sharing TCP", *ACM SIGCOMM Computer Communications Review*, vol. 28, no. 3, pp. 53–69, Jul. 1998.
29. **T. Hacker, B. Noble, B. Athey,** "Improving throughput and maintaining fairness using parallel TCP", in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.
30. **T. Hacker, P. Smith,** "Stochastic TCP: A statistical approach to congestion avoidance", in *Proc. PFLDnet 2008*, Manchester, Mar. 2008.
31. **F. Kuo, X. Fu,** "Probe-aided MulTCP: an aggregate congestion control mechanism", *ACM SIGCOMM Computer Communications Review*, vol. 38, no. 1, pp. 17–28, Jan. 2008.
32. **Eli Brosh, Salman Abdul Baset, Vishal Misra, Dan Rubenstein, Henning Schulzrinne,** "The delay-friendliness of TCP for real-time traffic", IEEE/ACM Transactions on Networking (TON) archive, Volume 18, Issue 5, October 2010
33. **B. Wang, J. Kurose, P. Shenoy, D. Towsley,** "A Model for TCP-based Video Streaming", ACM Multimedia Conference, Nov, 2004.
34. **B. Wang, J. Kurose, P. Shenoy, D. Towsley,** "Multimedia Streaming via TCP:

An Analytic Performance Study", ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 4, No. 2, Article 16, Publication date: May 2008
35. **T. Kim, M. H. Ammarb,** "Receiver Buffer Requirement for Video Streaming over TCP", Proceedings of Visual Communications and Image Processing Conference, San Jose, CA, 2006.
36. **T. Stockhammer,** "Dynamic Adaptive Streaming over HTTP – Design Principles and Standards", Proceedings of the second annual ACM conference on Multimedia systems, 2011.

CURRICULUM VITAE

Ime i prezime: **AFAN ČEČO**
Datum rođenja: **13. JUNI 1979.**
Mjesto rođenja: **Doboj, BiH**

Adresa stanovanja:
Fra Grge Martića 12
72 000 Zenica
Tel.: +387 32 406 306
E-mail:afan.ceco@etf.unsa.ba

Adresa zaposlenja:
BH Telecom d.d. Sarajevo
Masarykova 46, 72 000 Zenica
Tel.: +387 32 445 821
E-mail:afan.ceco@bhtelecom.ba

OBRAZOVANJE:

2005 - 2009: Postdiplomski studij, Elektrotehnički fakultet, Univerzitet u Sarajevu, BiH, stekao zvanje magistar elektrotehničkih nauka

1998 - 2004: Elektrotehnički fakultet, Univerzitet u Sarajevu, BiH, stekao zvanje dipl.ing.el. na Odsjeku za računarstvo i informatiku

1996 -1998: Tehnička škola Zenica, BiH

1995 -1996: Srednja škola Jean Baptiste Dumas, Ales, Francuska

OBJAVLJENI NAUČNI I STRUČNI RADOVI

1. „Performance Comparison of Active Queue Management Algorithms“, Afan Ceco, Novica Nosovic, Kenan Bradic, Telfor 2012 - Beograd (IEEExplore)
2. „Performance comparison of different TCP versions designed for networks with high speeds and over long distances“, Afan Ceco, Novica Nosovic, Mipro 2011 - Opatija (IEEExplore)
3. „Poboljšanje performansi TCP protokola pomoću mehanizama kontrole toka i zagušenja“, Afan Čečo, Časopis Telekomunikacije 9/29/2010 - Sarajevo.
4. „Poređenje performansi različitih verzija TCP protokola“, Afan Čečo, Kenan Bradić, Novica Nosović, Konferencija Telfor 2010 - Beograd.
5. „Analiza isplativosti uvođenja virtuelizacije servera u datacentar BH Telecom-a“, Kenan Bradić, Afan Čečo, Ilduza Husić, Konferencija Telfor 2010 - Beograd.

STRUČNO USAVRŠAVANJE

2008	Advanced VPNs, Juniper Networks, Istanbul, Turska
2008	Operation and Troubleshooting of Juniper Networks Routers, Juniper Networks, Istanbul, Turska
2008	Advanced Services Implementing Broadband Aggregation on Cisco Routers, Cisco Systems, Varšava, Poljska
2005	CISCO CCNA, instruktorske verzije, Fakultet za automatiku i informatiku, Politehnički univerzitet u Bukureštu, Rumunija
2003 - 2005:	CISCO CCNA, četiri semestra, Elektrotehnički fakultet u Sarajevu

ZAPOSLENJE:

2010 – 2013: Viši asistent na Univerzitetu Vitez u Travniku za predmete Računarske mreže, Napredne računarske mreže, kao i Zaštita podataka i računarskih sistema

2009 – trenutno: Sekretar Udruženja za promovisanje informatičke pismenosti „NET“ sa sjedištem u Zenici

2005 - trenutno: Stručni saradnik za paketske mreže u Direkciji Zenica, BH Telecom

2004 – trenutno: Instruktor na Cisco akademiji na ETF Sarajevo

2002 – 2004: Stručni saradnik za informacione sisteme, BS Telecom, Sarajevo

PROFESIONALNO ISKUSTVO:

Trenutno zaposlen kao Stručni saradnik za paketske mreže u BH Telecomu. U svom radu učestvovao u većem broju projekata izgradnje i proširenja paketske mreže BH Telecoma, od kojih se ističu realizacija mreže MUX-ova (leased line pristup), DSLAM-ova (ADSL pristup), HotSpot-ova (wireless pristup), optičkog pristupa, itd. Pri tome radio na izradi projekata izgradnje i proširenja mrežnih kapaciteta, dokumentacije za nabavku opreme, učestvovao u implementaciji, kao i bio član komisija za tehnički prijem i puštanje u rad opreme. Osim toga, treba spomenuti i svakodnevne poslove na održavanju i eksploataciji gore spomenutih sistema i mreža.

DODATNE NAPOMENE:

- Odlično poznavanje engleskog i francuskog jezika u govornoj i pisanoj formi.
- Na Elektrotehničkom fakultetu u Sarajevu uradio diplomski rad na temu TCP/IP protokola, kao i magistarski rad na temu: ”Načini poboljšanja performansi Transmisionog kontrolnog protokola pomoću mehanizama kontrole toka i zagušenja”.