Poređenje performansi različitih verzija TCP protokola

Afan Čečo, Kenan Bradić, Novica Nosović

***Sadržaj* — 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. U radu su prikazani rezultati poređenja performansi različitih verzija TCP protokola, koji su dobiveni pomoću simulacije na OPNET simulatoru. Rad dokazuje kako se mogu postići znatna poboljšanja performansi mreže upotrebom mehanizama kontrole zagušenja na transportnom sloju.**

Ključne riječi — brza retransmisija, brzi oporavak, izbjegavanje zagušenja, New Reno, Reno, SACK, spori početak, Tahoe, TCP.

I. UVOD

TCP/IP (Transmission Control Protocol / Internet Protocol) je standardni skup protokola pomoću kojeg radi Internet. TCP protokol je, pored IP-a, jedan od dva glavna protokola u sklopu TCP/IP skupa protokola. TCP protokol omogućava pouzdan transfer konekcionog tipa. Prije transmisije podataka, dva komunicirajuća hosta prolaze kroz proces sinhronizacije da bi uspostavili virtuelnu konekciju. Ovakav sinhronizacioni proces osigurava da i jedna i druga strana budu spremne za transmisiju podataka, te omogućava uređajima da odrede inicijalne sekvencijalne brojeve. Spomenuti proces je poznat kao „three-way handshake“. To je proces iz tri koraka koji uspostavlja virtuelnu konekciju između dva uređaja.

Prvobitni TCP bi započeo konekciju tako što bi predajnik ubacio višestruke segmente u mrežu, sve do veličine prozora objavljene od strane prijemnika (eng. advertised window). Pored činjenice što je navedeno prihvatljivo kada su dva hosta na istom LAN-u (eng. Local Area Network – lokalna mreža), mogu se pojaviti problemi u situacijama kada postoje ruteri i sporiji linkovi između predajnika i prijemnika. Pojedini posrednički ruteri moraju stvarati redove čekanja sa paketima, tako da je moguće da navedeni ruter ostane bez slobodnog prostora za privremeno smještanje paketa. Algoritam koji služi da se izbjegne gore opisani problem se naziva Slow-Start (spori početak). Radi tako što posmatra da li brzina kojom se novi paketi ubacuju u mrežu odgovara brzini kojom se potvrde prijema vraćaju od druge strane.

Zagušenje se može desiti kada podaci stignu sa brzog LAN-a i budu poslati na sporiji WAN (eng. Wide Area Network – mreža koja pokriva šire geografsko područje). Zagušenje se takođe može desiti kada višestruki dolazni protoci stižu na ruter čiji je izlazni kapacitet manji nego što iznosi suma svih ulaza. Congestion Avoidance (izbjegavanje zagušenja) je način da se prevaziđe problem gubljenja paketa.

Fast Retransmit (brza retransmisija) je poboljšanje TCP-a koje smanjuje vrijeme koje predajnik čeka prije retransmisije izgubljenog segmenta.

Kada Fast Retransmit pošalje ono što bi trebalo biti nestali segment, primjenjuje se Congestion Avoidance, a ne Slow Start. To se naziva Fast Recovery (brzi oporavak) algoritam.

II. VARIJANTE TCP PROTOKOLA

A. TCP Tahoe

TCP se zasniva na principu "očuvanja paketa" (eng. conservation of packets): što znači da ako konekcija radi na raspoloživom kapacitetu propusnog opsega, onda se paket ne ubacuje u mrežu sve dok drugi paket ne napusti mrežu. TCP implementira navedeni princip tako što koristi potvrde (eng. acknowledgment, ACK) da tempira odlazeće pakete, zbog toga što potvrda znači da je određeni paket napustio mrežu. Isto tako, pored objavljenog prozora, TCP održava prozor zagušenja (eng. congestion window, cwnd) da bi predstavio mrežni kapacitet [1]. Predajnik može transmitovati sve do minimuma između vrijednosti prozora zagušenja i objavljenog prozora. Prozor zagušenja je kontrola toka koju nameće predajnik, dok je objavljeni prozor kontrola toka koju nameće prijemnik. Prva kontrola je zasnovana na predajnikovoj percepciji mrežnog zagušenja, dok je druga vezana za veličinu raspoloživog prostora u baferu na prijemniku za datu konekciju.

Transmisije paketa kod TCP-a su tempirane dolazećim potvrdama. Međutim, postoji problem kada se konekcija prvi put uspostavlja, jer da bi imali potvrde moramo imati podatke u mreži, a da bi ubacili podatke u mrežu moramo imati potvrde. Da bi se prevazišao opisani problem, Tahoe 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 svaki primljeni ACK povećava se cwnd za 1, tako da u prvom RTT-u (eng. round-trip time - vrijeme

Afan Čečo, Elektrotehnički fakultet u Sarajevu, Zmaja od Bosne bb, 71000 Sarajevo, BiH (telefon 387-61-756041, e-mail: afan.ceco@etf.unsa.ba)

Kenan Bradić, BH Telecom d.d. Sarajevo, Obala Kulina bana 8, 71000 Sarajevo, BiH (telefon: 387-30-519829, e-mail: kenan.bradic@bhtelecom.ba)

Novica Nosović, Elektrotehnički fakultet u Sarajevu, Zmaja od Bosne bb, 71000 Sarajevo, BiH (telefon 387-33-250725, e-mail: nnosovic@etf.unsa.ba)

povratnog puta) šaljemo 1 paket, u drugom šaljemo 2, a u trećem šaljemo 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 detektuje gubitke paketa zahvaljujući vremenskim ograničenjima (eng. timeout), koja ukazuju da je istekao vremenski interval unutar kojeg je trebala stići potvrda. Kod običnih implementacija, imamo 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.

Za Congestion Avoidance Tahoe koristi "dodatna povećanja i višestruka smanjenja" (eng. Additive Increase Multiplicative Decrease, AIMD). Gubitak paketa se uzima kao znak zagušenja i Tahoe 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.

Tahoe 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, 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.

B. TCP Reno

Reno zadržava osnovne principe Tahoe algoritma, kao što su Slow Start i dugotrajni retransmisijski tajmer (eng. timer). Međutim, Reno dodaje više inteligencije tako što ranije detektuje izgubljene pakete i protočna struktura se ne prazni svaki put kada se izgubi paket. Reno 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 primimo 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 sugeriše algoritam koji se naziva Fast Retransmit. 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 čini je da poslije gubitka paketa ne smanjuje prozor zagušenja na 1. Razlog je što bi to ispraznilo protočnu strukturu. Reno ulazi u algoritam koji se naziva Fast Retransmit [2]. 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.

Reno se ponaša veoma dobro kada se radi o malim gubicima paketa. Ali, kada imamo višestruke gubitke paketa u istom prozoru, onda se Reno ne ponaša previše dobro i njegove performanse su približno iste kao kod TCP Tahoe u istim uslovima. Razlog za to je što može detektovati samo pojedinačne gubitke paketa. Ako postoje višestruki gubici paketa, onda prva informacija o gubitku paketa dolazi kada se prime duplicirani 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.

C. TCP New Reno

TCP New Reno predstavlja manju modifikaciju urađenu u odnosu na TCP Reno. U stanju je da otkrije višestruke gubitke paketa, tako da je mnogo efikasniji nego Reno u slučaju višestrukih gubitaka paketa. Kao i Reno, New Reno takođe ulazi u Fast Retransmit kada primi višestruke dupliciranе pakete, ali se razlikuje od TCP Reno 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 sa smanjivanjem cwnd prozora više puta.

Fast Retransmit faza je ista kao kod TCP Reno. Razlika je u Fast Recovery fazi koja dozvoljava višestruke retransmisije. Svaki put kada New Reno uđe u Fast Recovery, bilježi maksimalni preostali segment. Kada stigne 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;
- Ako ACK predstavlja samo parcijalni ACK, onda New Reno zaključuje da je sljedeći segment u nizu izgubljen i retransmituje segment, te postavlja broj primljenih dupliciranih ACK-ova na nula. New Reno izlazi iz Fast Recovery-a tek kada su svi podaci u prozoru potvrđeni [3].

New Reno pati od činjenice 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.

D. TCP SACK

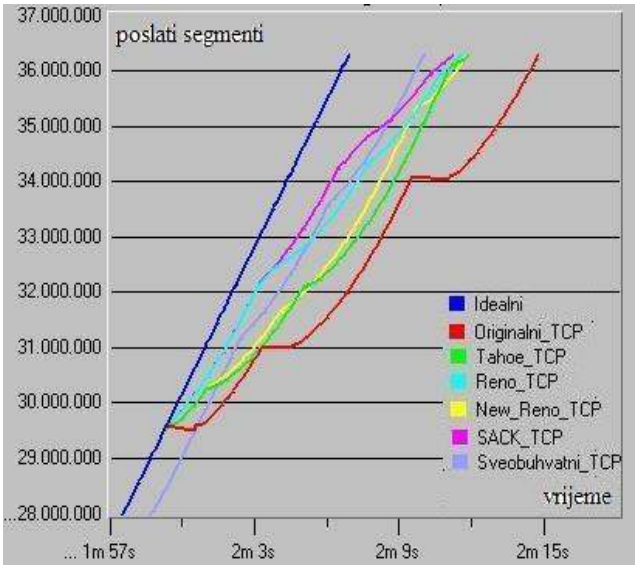
TCP sa "selektivnim potvrdama" (eng. Selective Acknowledgments, SACK) je proširenje u odnosu na TCP Reno i radi na problemima sa kojima se susreću TCP Reno i TCP New Reno, a posebno na detekciji višestrukih izgubljenih paketa, kao i retransmisiji više od jednog izgubljenog paketa po RTT-u.

SACK zadržava Slow Start i Fast Retransmit dijelove od TCP Reno. Isto tako, ima dugotrajna vremenska ograničenja kao i Tahoe, 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. Tako da 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 provjerava koji segmenti nisu primljeni i ponovo ih otprema. Ako nema takvih preostalih segmenata, onda šalje novi paket [4]. Tako se postiže da više od jednog izgubljenog segmenta može biti poslato u jednom RTT-u.

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

III. SIMULACIJA TCP PROTOKOLA

Simulacija je urađena na simulatoru OPNET IT Guru Academic Edition. Sastoji se od sedam različitih scenarija, od kojih svaki predstavlja različite mehanizme kontrole toka i zagušenja. Predstavljeni scenariji su sljedeći: Idealni TCP, Originalni TCP, Tahoe TCP, Reno TCP, New Reno TCP, SACK TCP i Sveobuhvatni TCP. Idealni TCP je uveden za potrebe simulacije i predstavlja idealnu situaciju, bez grešaka na prenosnom putu. Originalni TCP predstavlja prvobitnu verziju TCP protokola, bez dodatnih mehanizama kontrole toka i zagušenja. Tahoe TCP, Reno TCP, New Reno TCP i SACK TCP su standardne verzije TCP protokola. Sveobuhvatni TCP (eng. Full-Featured TCP) je uveden za potrebe simulacije i predstavlja situaciju kada bi se koristile sve trenutno raspoložive opcije za kontrolu toka i zagušenja. Pod tim opcijama se podrazumijeva: skaliranje prozora (eng. Window Scaling), selektivni ACK-ovi (SACK), eksplicitna notifikacija zagušenja (eng. Explicit Congestion Notification, ECN), Nagle-ov algoritam, Karn-ov algoritam, kao i vremenske oznake (eng. Timestamp). U osnovi Sveobuhvatnog TCP-a se nalazi TCP Reno, ali sa uključenim svim dodatnim opcijama.

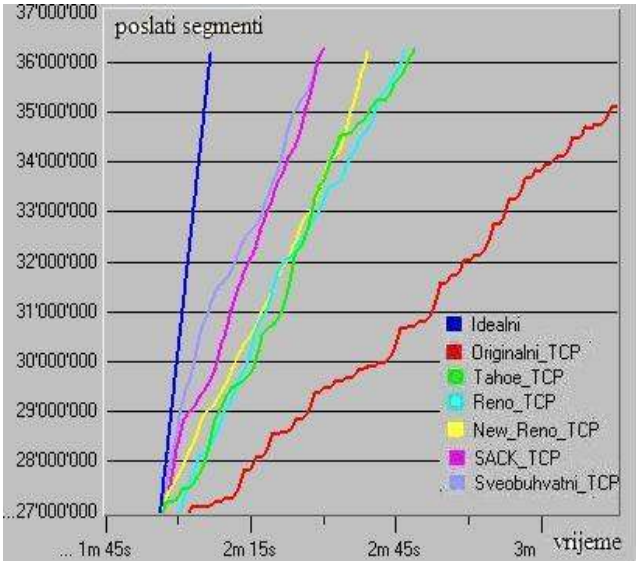


Sl. 1. Poređenje TCP verzija pri 0.05 % grešaka

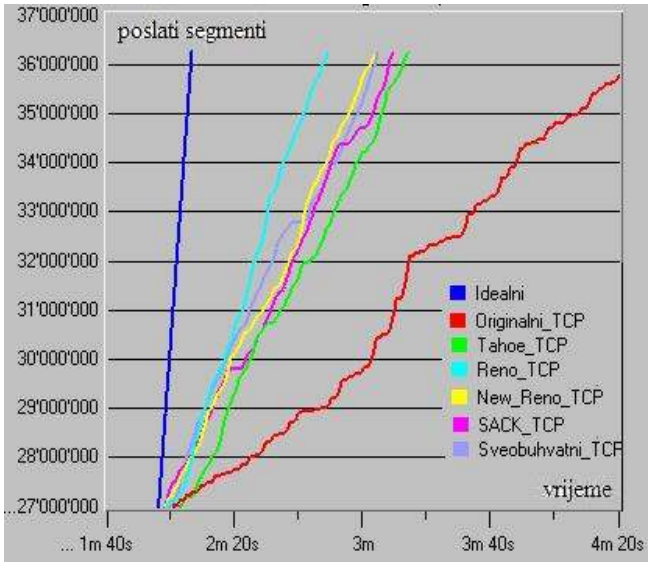
Greške su skalirane tako da prvo iznose 0,05 %, zatim 0,5 %, pa 1 % i na kraju 5 %. Kada greške iznose 0,05 %, Sveobuhvatni TCP ostvaruje najbolje rezultate, ispred SACK TCP-a, Reno TCP-a, New Reno TCP-a, Tahoe TCP-a i Originalnog TCP-a (Sl. 1).

Ovo je očekivani poredak, s obzirom na kompleksnost algoritama, osim što New Reno TCP nije pokazao svoju prednost u odnosu na Reno TCP. Treba napomenuti da se radi o manjim greškama, tako da prednosti New Reno TCP-a nisu došle do izražaja. Može se uočiti da Sveobuhvatni TCP ima performanse koje su najbliže Idealnom TCP-u, naročito što vrijeme više odmiče. To znači da Sveobuhvatni TCP uspjeva poslati više segmenata podataka (veću količinu podataka, izraženo u bajtima) u istom vremenskom intervalu od drugih verzija TCP-a.

Kada se greške povećaju na 0,5 %, Sveobuhvatni TCP bilježi i dalje najbolje performanse, a nakon njega ponovo slijedi SACK TCP (Sl. 2). New Reno TCP sada pokazuje svoju prednost u odnosu na Reno TCP. Najlošije rezultate opet pokazuju Tahoe TCP i Originalni TCP.



Sl. 2. Poređenje TCP verzija pri 0.5 % grešaka



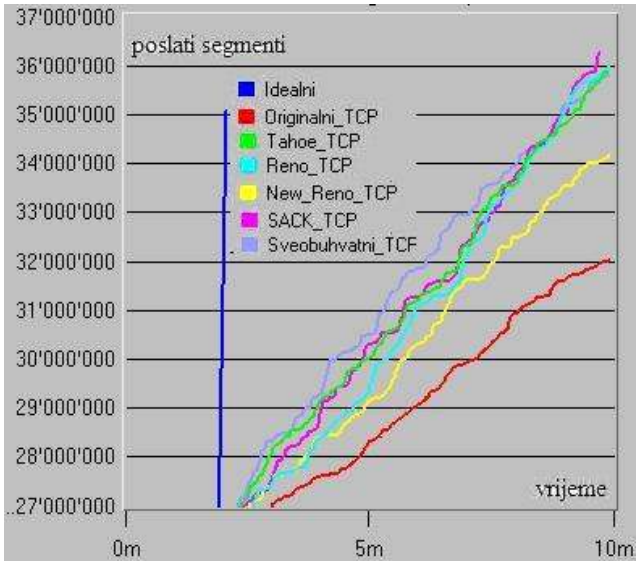
Sl. 3. Poređenje TCP verzija pri 1 % grešaka

Kada se greške povećaju na 1 %, Reno TCP se pokazuje kao najbolji, a poslije njega dolazi New Reno TCP (Sl. 3). Zatim, slijede Sveobuhvatni TCP, SACK TCP, Tahoe TCP i Originalni TCP. U ovom slučaju, Reno i New Reno TCP su se pokazali kao bolji od ostalih.

Pri greškama od 5 %, SACK TCP bilježi najbolje rezultate (Sl. 4). Zatim, slijede Reno TCP, Tahoe TCP, Sveobuhvatni TCP, New Reno TCP i Originalni TCP. Pri veoma izraženim greškama, Sveobuhvatni TCP se pokazao kao loše rješenje.

Veći procenti grešaka (> 0,5 %) se mogu pojaviti prvenstveno na bežičnim medijima, ali onda postoje protokoli nižeg nivoa koji koriguju greške i dovode ih na istu vrijednost kao kod drugih medija.

Na osnovu postignutih rezultata simulacije može se načiniti poredak svih predočenih verzija TCP protokola, od verzije sa najboljim rezultatima do one sa najlošijim. Poredak različitih verzija TCP protokola pri skaliranim greškama je prikazan u Tabeli 1.



Sl. 4. Poređenje TCP verzija pri 5 % grešaka

TABELA 1. POREDAK RAZLIČITIH TCP VERZIJA

	0.05 % greš.	*0.5 % grešaka*	*1 % grešaka*	*5 % grešaka*
1	Sveob. TCP	Sveob. TCP	Reno TCP	SACK TCP
2	SACK TCP	SACK TCP	N. Reno TCP	Reno TCP
3	Reno TCP	N. Reno TCP	Sveob. TCP	Tahoe TCP
4	N. Reno TCP	Reno TCP	SACK TCP	Sveob. TCP
5	Tahoe TCP	Tahoe TCP	Tahoe TCP	N. Reno TCP
6	Original. TCP	Original. TCP	Original. TCP	Origin. TCP

IV. ZAKLJUČAK

Na osnovu rezultata simulacije može se zaključiti da SACK TCP ostvaruje iste, najbolje prosječne rezultate, kao Sveobuhvatni TCP. Nakon toga, slijedi Reno TCP na drugom mjestu. Zatim, New Reno TCP zauzima treće mjesto. Tahoe TCP ostvaruje četvrto mjesto. Originalni TCP se nalazi na zadnjem mjestu u svakom od testiranih slučajeva.

Izvedena simulacija je pokazala da nijedna od predočenih verzija TCP protokola ne može ostvariti najbolje rezultate u svim okolnostima. Pri tome, treba napomenuti da greške na realnom prenosnom putu obično iznose manje od 0,5 %.

Rad dokazuje da je dobro rješenje koristiti TCP sa selektivnim potvrdama (SACK), ali isto tako i sa drugim opcijama (ECN, Window Scaling, Timestamp, Nagle, Karn).

Može se očekivati, da će se u budućnosti, istraživanja nastaviti u smjeru pronalaska novih mehanizama i stvaranja novih verzija TCP protokola. Pri tome, naročitu perspektivu ima kombinovanje postojećih verzija TCP protokola u zajedničku implementaciju, koja bi imala superiorne performanse.

LITERATURA

[1] V. Jacobson, "Congestion Avoidance and Control," SIGCOMM Symposium on Communication Architectures & Protocols, 1988.
[2] V. Jacobson, "Modified TCP Congestion Control and Avoidance Algorithms," Technical Report 30, Apr 1990.
[3] S. Floyd, T. Henderson, "The New-Reno Modification to TCP's Fast Recovery Algorithm," IETF RFC 3782.
[4] K. Fall, S. Floyd, "Simulation Based Comparison of Tahoe, Reno and SACK TCP," Computer Communications Review, 1996.

ABSTRACT

TCP/IP protocol is the standard of the Internet, which is the largest computer network in the world. IP protocol and IP routing, enables forwarding of packets between networks. However, IP does not guarantee that data packets will be delivered. For reliable transport and regulation of data flow from the source to the destination, the TCP protocol is responsible. This paper presents the results of comparing the performance of different versions of TCP protocol, which are obtained by simulation in the OPNET simulator. The paper shows how significant network performance improvements can be achieved, by using the congestion control mechanisms at the transport layer.

PERFORMANCE COMPARISON OF DIFFERENT TCP VERSIONS

Afan Čečo, Kenan Bradić, Novica Nosović