https://doi.org/10.37528/FTTE/9788673954899/POSTEL.2024.13

PRIMENA TEHNIKE OBLIKOVANJA SAOBRAĆAJA U BACKUP PROCESIMA

Slobodan Mitrović, Valentina Radojičić
Univerzitet u Beogradu - Saobraćajni fakultet
s.mitrovic@sf.bg.ac.rs, valentin@sf.bg.ac.rs

Rezime: *Upravljanje lokalnim mrežnim saobraćajem često se oslanja na primenu odgovarajućih tehnika sa zadatkom kontrolisane upotrebe raspoloživih mrežnih kapaciteta u prisustvu konkurentnih mrežnih aktivnosti kao što su backup procesi. U ovom radu predstavljena je uporedna analiza primene algoritama za upravljanje paketskim redovima za čekanje, u okviru paketskog filtera za oblikovanje saobraćaja. Analizom je obuhvaćeno nekoliko algoritama koji su široko primenjeni u praksi, uključujući i algoritam za ravnomerno tretiranje paketa u redu za čekanje sa kontrolisanim kašnjenjem (Fair Queuing with Controlled Delay – FQ-CoDel). Cilj ove uporedne analize je da se ispita pogodnost upotrebe navedenih algoritama za potrebe realizacije backup procesa u lokalnoj računarskoj mreži.*

Ključne reči: *oblikovanje saobraćaja, backup proces, paketski redovi, FQ-CoDel*

1. Uvod

Struktura savremenih servisa, koji su u upotrebi u lokalnim računarskim mrežama (LAN), u potpunosti reflektuje multifunkcionalnost potreba korisnika u pogledu pristupa podacima, multimedijalnim sadržajima, kao i načinu komunikacije. U skladu sa tim, lokalno serversko okruženje vrlo često ima kompleksnu strukturu, čiji je zadatak da na efikasan način realizuje interne servise koristeći raspoložive mrežne resurse. Imajući u vidu važnost obrađenih i razmenjenih informacija, proces formiranja i upravljanja rezervnim kopijama serverskih podataka, (odnosno tzv. *backup* proces) može se smatrati ključnom merom zaštite poslovnih i drugih procesa u okviru posmatranih institucija, gde rezervne kopije služe kao prva linija odbrane od gubitka podataka usled slučajnog brisanja, sajber-napada, zagušenja, kvarova na hardveru ili katastrofalnih događaja. Proces formiranja rezervnih kopija, po pravilu predstavlja upravljivu seriju realizacija zadataka koji se odnose na kopiranje i skladištenje podataka sa primarnih namenskih servera na tzv. *backup* servere. U zavisnosti od operativne uloge u poslovnim procesima, pojedini servisi mogu zahtevati različite prioritete prenosa podataka i organizaciju sistema skladištenja, zbog čega konfiguracije formiranja rezervnih kopija direktno utiču na trajanje, pouzdanost i ukupan uspeh procesa pravljenja rezervnih kopija. Na ovaj način, *backup* proces može imati određeni uticaj na performanse lokalne računarske

mreže tokom i/ili van radnog vremena, u zavisnosti od definisane konfiguracije, na taj način što može izazvati pojavu kašnjenja u prenosu podataka drugih namena, kao i usporavanje ostalih poslovnih i drugih operacija čija se efikasnost oslanja na određen nivo raspoloživih resursa posmatrane LAN mreže u datom trenutku. Zbog toga, uvođenje mehanizama upravljanja saobraćajem predstavlja neophodan korak koji vodi ka efikasnijoj upotrebi raspoloživih resursa na posmatranoj LAN mreži.

Imajući navedeno u vidu, u ovom radu biće predstavljena uporedna analiza primene algoritama za upravljanje paketskim redovima za čekanje, u slučaju realizacije *backup* procesa u lokalnoj računarskoj mreži. Uporedna analiza će obuhvatiti nekoliko algoritama koji su široko primenjeni u praksi (odnosno u komercijalno dostupnim mrežnim uređajima), uključujući i algoritam za ravnomerno tretiranje paketa u redu za čekanje sa kontrolisanim kašnjenjem (*Fair Queuing with Controlled Delay* – FQ-CoDel). Ispitivani algoritmi su pozicionirani unutar mehanizma za primenu saobraćajnih polisa ili za oblikovanje saobraćaja baziranom na primeni tzv. algoritma klasične (jednostruke) korpe sa žetonima (*Token Bucket algorithm* - TB). Prikazani eksperiment je osmišljen sa ciljem da evaluira efikasnost primene navedenih algoritama u uslovima simulacije prenosa *backup* podataka u prisustvu različitih tokova podataka za druge namene.

Ostatak rada je struktuiran na sledeći način: u drugom poglavlju prikazan je pregled najvažnijih elemenata *backup* procesa, kao i značaj primene mehanizama kontrole mrežnog saobraćaja prilikom realizacije ove vrste serverskih aktivnosti. U trećem poglavlju prikazan je pregled algoritama za upravljanje paketskim redovima za čekanje. Četvrto poglavlje sadrži opis sprovedenog eksperimenta, kao i prikaz dobijenih rezultata, sa odgovarajućim zaključcima na kraju rada.

2. *Backup* proces i kontrola saobraćaja

Backup proces, odnosno proces pravljenja rezervnih kopija podataka, može biti organizovan na nivou svakog pojedinačnog servera ili na grupi servera, koji svi mogu biti u okviru iste LAN mreže ili podeljeno – u okviru LAN mreže i u WAN okruženju (u okviru definisanih *Disaster Recovery* centara ili drugih lokacija te namene). Takođe nije redak slučaj da se *backup* proces odvija i u kombinovanim režimima. U okviru posmatrane LAN mreže, u ovom procesu po pravilu učestvuje veći broj servera, koji svoje sadržaje kopiraju na jedan ili nekoliko namenskih *backup* servera. Serveri koji treba da kopiraju svoje sadržine, mogu biti namenski (*dedicated*) serveri ili serveri koji hostuju virtuelizovane mašine, odnosno druge virtuelizovane servere. U tom smislu, rezervne kopije podataka mogu se pojaviti u obliku [1]:

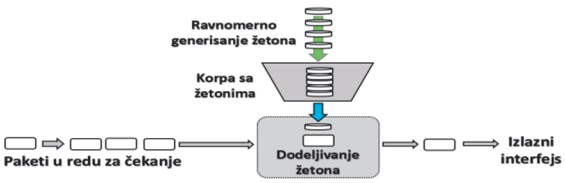
- *celokupnih* rezervnih kopija (*full backup*), koje predstavljaju kopije celokupnih sadržaja posmatranih medija na kojima su uskladišteni originalni podaci. *Celokupne* rezervne kopije mogu se kroz mrežu prenositi u originalnom ili kompresovanom režimu ka odgovarajućoj lokaciji na *backup* serveru. Ova vrsta kopije omogućava jednostavnu rekonstrukciju originalnih sadržaja, ali zahteva veći skladišni prostor na *backup* serveru, kao i duže vreme za prenos kopije kroz mrežu;
- *bare metal* kopija, koje u jednoj datoteci sadrže kompletnu strukturu podataka jedne skladišne jedinice (cele particije ili *hard drive*) originalnog servera. Ovakva struktura može uključivati i kompletan operativni sistem, koji se može koristiti i za potrebe eventualne rekonstrukcije na praznom medijumu u slučaju većih oštećenja originalne infrastrukture;

- *inkrementalnih* rezervnih kopija, koje sadrže samo one promene koje su napravljene u periodu od formiranja poslednje rezervne kopije (bilo da se radi inicijalno formiranoj celokupnoj kopiji ili prethodno formiranoj inkrementalnoj kopiji). Primene inkrementalnog principa podrazumeva mogućnost da se vrši pristup uskladištenim kopijama na *backup* serveru radi identifikacije razlika. Proces formiranja inkrementalnih kopija može biti značajno brži u poređenju sa procesom formiranja celokupnih rezervnih kopija, ali zahteva više vremena za rekonstrukciju sadržaja na originalnim skladišnim lokacijama;
- *diferencijalnih* rezervnih kopija, koje sadrže one promene koje su napravljene u periodu od formiranja celokupne kopije. Proces formiranja diferencijalnih kopija zahteva više vremena u poređenju sa slučajem formiranja inkrementalnih kopija, ali sa druge strane zahteva manje vremena potrebnog za rekonstrukciju sadržaja na originalnim skladišnim lokacijama.

Osim navedenih koncepata formiranja rezervnih kopija postoji i koncept formiranja tzv. brzih snimaka (*snapshots*). *Snapshot*-ovi predstavljaju zamrznutu sliku svih podataka u vreme pravljenja rezervne kopije, zbog čega su pogodni za primenu u slučajevima kada treba izvršiti trenutni oporavak u slučaju gubitka podataka, oštećenja ili kvarova sistema [2]. Formiranje *snapshot*-ova je korisno u okruženjima kao što su virtuelne mašine gde stanja servera mogu varirati u zavisnosti od aktivnosti. Ipak, primena *Snapshot* koncepta ima i određene nedostatke. Naime, *snapshot*-ovi se obično čuvaju na istim serverima koji hostuju i virtuelne mašine, tako da u slučaju ugroženosti *host* servera mogu biti ugroženi i *snapshot*-ovi koji su uskladišteni na istoj lokaciji. Takođe, ukoliko server formira previše s*napshot*-ova, performansa njegovog rada može opasti tokom vremena eksploatacije. Važno je napomenuti i da *snapshot*-ovi sami po sebi ne predstavljaju datoteke koje se mogu *ad-hoc* koristiti za čuvanje podataka u backup procesu (kao što je to slučaj sa *bare metal* kopijama), imajući u vidu da su *snapshot*-ovi inherentno zavisni od korespondentnog virtuelnog diska pripadajuće virtuelne mašine. Ipak, savremena rešenja za formiranje rezervnih kopija virtuelnih diskova mogu da formiraju upotrebljive kopije bez gašenja virtuelnih mašina na osnovu stanja *snapshot*-ova primenom posebnih mehanizama replikacije na alternativnu skladišnu lokaciju [3]. Tek nakon završetka tog procesa, rezultujuće *backup* datoteke mogu se preneti u skladišni prostor određenog *backup* servera.

Imajući navedeno u vidu, savremeni pristupi vezani za strategije formiranja rezervnih kopija imaju hibridni karakter, kojima se kombinuje formiranje svih navedenih oblika rezervnih kopija, kao i *snaphot*-ova zavisno od namene. Raspoloživost kapaciteta odgovarajućih mrežnih resursa ima važnu ulogu u realizaciji efikasnog *backup* procesa. Tome u najvećoj meri doprinosi činjenica da su TCP protokoli, koji po pravilu služe za prenos podataka, konekciono orjentisani i po svojoj prirodi skloni zauzimanju svog raspoloživog kapaciteta posmatrane veze koji im stoji na raspolaganju. Na ovaj način TCP protokoli doprinose formiranju tzv. *bursty* karaktera mrežnog saobraćaja kojim se dovodi do generisanja veoma velikog broja paketa u jedinici vremena. Zbog toga može da se desi da procesi formiranja i prenosa rezervnih kopija kroz mrežu utiču na pad performansi mreže i na taj način uspore ili čak ugroze druge poslovne i ostale procese koje se oslanjaju na određen nivo raspoloživosti kapaciteta mrežnih resursa. Da bi se u prisustvu aktivnih *backup* procesa obezbedila i realizacija drugih aktivnosti koje se oslanjaju na obradu podataka u realnom vremenu koristeći resurse posmatrane LAN mreže, potrebno je uvesti mehanizme upravljanja mrežnim saobraćajem. Ovo se postiže

definisanjem određenih tipova mrežnog saobraćaja kojima se dodeljuju odgovarajući nivoi prioriteta radi realizuje kritičnih poslovnih funkcija. Upravljanje lokalnim mrežnim saobraćajem se pored primene virtuelnih LAN mreža (odnosno, VLAN-ova na *Data-Link* sloju) namenskog rutiranja saobraćaja (na mrežnom sloju), često oslanja na primenu tehnika saobraćajnih polisa (*Traffic Policing*), kao i tehnika oblikovanja saobraćaja (*Traffic Shaping*). Obe tehnike pružaju mogućnost kontrolisane upotrebe raspoloživih mrežnih kapaciteta u prisustvu konkurentnih mrežnih aktivnosti, na taj način što će limitirati *bursty* karakter mrežnog saobraćaja na nivo koji je neophodan za realizaciju posmatranih procesa u celini. U ovom slučaju konkurentnost se upravo javlja između *backup* procesa sa jedne strane i drugih vrsta mrežnih aktivnosti sa druge, pri čemu svi procesi konkurišu za što veći udeo u zauzimanju tzv. tehničkog (tj. realnog) propusnog opsega linka (*Access Information Rate* - AIR). S druge strane, limitiranje pojedinih mrežnih procesa u pogledu upotrebe AIR, sprovodi se tako da se koristi definisani udeo propusnog opsega (*Committed Information Rate* - CIR) [4], kojim se garantuje mogućnost realizacije u definisanom vremenu. Limitiranje resursa sprovodi se upotrebom kontrole protoka paketa tako što paketi mogu ulaziti u paketski filter bilo kojim intenzitetom, ali je učestanost njihovog izlaza iz filtera određena dužinom reda za čekanje, kao i načinom na koji se „odobrava“ njihov izlazak iz posmatranog paketskog filtera. Dakle, algoritmi upravljanja redom za čekanje i algoritmi „odobravanja“ izlaska paketa se razlikuju po svojoj ulozi i mogu imati različiti uticaj na efikasnost rada paketskog filtera u zavisnosti od vrste primenjenih algoritama. U ovako koncipiranim paketskim filterima, odobravanje izlaska paketa se često realizuje primenom algoritma tzv. *korpe žetona* (*Token bucket* - TB) [5]. Ovaj algoritam je zasnovan na metafori tzv. *korpe* koja se puni *žetonima* tokom vremena, pri čemu nadalje svaki *žeton* „ispada“ iz *korpe* i dodeljuje se ili definisanoj jediničnoj količini podataka (u bajtovima/bitima) ili se dodeljuje samom paketu. *Žeton* predstavlja vrstu dozvole da posmatrani paket može napustiti paketski filter ka definisanom izlaznom interfejsu (Slika 1). Princip rada *korpe sa žetonima*, zasnovan je na sledećim pravilima: *žetoni* se uvek generišu ravnomerno, u istim vremenskim intervalima. Nadalje, *korpa* ima limitiran kapacitet, odnosno broj *žetona* koji može stati u korpu je ograničen.



Slika 1. Princip rada korpe sa žetonima

Kada *korpa* biva u potpunosti napunjena žetonima, svaki naknadno generisani žeton može se odbaciti ili se može prebaciti u drugu korpu, u slučaju kompleksnijih algoritama. To znači da se, u nekoj jedinici vremena, na izlasku iz TB paketskog filtera može pojaviti onoliko paketa koliko ima žetona u korpi. Na taj način kapacitet korpe definiše veličinu *burst*-a, koji se može pojaviti na izlasku iz filtera. S druge strane, ukoliko tokom nailaska paketa ponestane *žetona* u *korpi*, paket će biti tretiran u zavisnosti od tehnike koja je primenjena u tom slučaju:

- ukoliko je primenjena tehnika saobraćajnih polisa, posmatrani paket će biti odbačen;

- ukoliko je primenjena tehnika oblikovanja saobraćaja, paket će biti „stavljen na čekanje" da se nova količina *žetona* generiše i smesti u *korpu*, dok u slučaju kompleksnijih algoritama može biti „preusmeren" na sekundarnu *korpu* koja može biti u ravnopravnom ili hijerarhijskom položaju [6, 7] u odnosu na primarnu *korpu* [8]. Pored navedenog, u TB paketskim filterima, značajnu ulogu imaju i redovi za čekanje paketa. Način na koji se paketi iz reda za čekanje mogu slati na pridruživanje sa žetonom, može u velikoj meri uticati na regulaciju tokova saobraćaja, koji se generišu u multikorisničkom okruženju uz pridržavanje zadatih CIR ograničenja. Shodno navedenom, u narednom poglavlju će biti predstavljen kratak pregled onih algoritama za upravljanje paketima u redovima za čekanje, koji su široko zastupljeni u komercijalno dostupnim uređajima za rutiranje mrežnog saobraćaja.

3. Pregled algoritama za formiranje paketskih redova za čekanje

Paketski red za čekanje ima ulogu privremenog bafera gde se dolazni paketi prihvataju i čuvaju po odgovarajućem setu pravila, odnosno algoritmu pre nego što bivaju poslati na pridruživanje sa žetonima. Paketski red za čekanje ima dužinu koja se podešava u skladu sa primenjenim algoritmom za opsluživanje paketa. Ukoliko nadolazeći paketi premaše definisani prag maksimalno dozvoljenog intenziteta saobraćaja, nadolazeći paketi mogu biti odbačeni (tehnika saobraćajne polise) ili stavljeni na čuvanje (tehnika oblikovanja saobraćaja). U tom slučaju, nadolazeći paketi će biti zadržani u onom broju koji odgovara definisanoj dužini reda, dok će prekobrojni paketi biti odbačeni (tzv. *tail-drop* efekat) [9]. Način na koji se rukuje paketima unutar reda je regulisan posebnim algoritmima, među kojima se izdvajaju tipski algoritmi (slika 2):

- algoritam *First-In-First-Out* (FIFO) je osnovni algoritam čekanja u redu gde se paketi propuštaju tačno onim redosledom kojim su i pristigli. U FIFO algoritmu ne postoji prioritet ili kategorizacija paketa, odnosno svaki paket čeka u redu onoliko koliko je potrebno da se svakom od paketa dodeli po jedan žeton, na putu ka izlazu iz paketskog filtera. Iako navedene karakteristike čine FIFO algoritam veoma brzim, podjednak tretman svakog paketa umanjuje adekvatnost primene ovog algoritma. To je posebno uočljivo u slučajevima kada treba klasifikovati dolazni saobraćaj po nekoj vrsti prioriteta, jer može doći do kašnjenja visoko priritetnog saobraćaja u slučajevima kada je red za čekanje popunjen paketima koji su nižeg prioriteta [10];
- SFQ (*Stochastic Fairness Qeuing*) algoritam je jedan od algoritama koji su osmišljeni da otklone nedostatak prioritetizacije paketa, sa ciljem da se raspoloživi kapacitet pravično raspodeli u odnosu na posmatrane tokove podataka. U tom smislu, SFQ opslužuje dolazni saobraćaj na taj način što upotrebom funkcije za heširanje klasifikuje dolazne pakete na osnovu određenog broja identifikatora i smešta ih u određen broj FIFO bafera (obično 1024), gde svaki od njih ima podjednake šanse da bude opslužen u daljem procesu. Paketi koji se nalaze na izlazima iz FIFO grupe bafera, nadalje se preuzimaju – po jedan iz svakog bafera u cikličnom (*Round-Robin*) maniru i smeštaju u red za dodeljivanje žetona na putu ka izlazu iz paketskog filtera [10]. Iako je princip pravičnosti pri izboru tokova saobraćaja izraženiji u poređenju sa FIFO algoritmom, nedostatak SFQ algoritma se ogleda u pravičnosti izbora saobraćajnih tokova u odnosu na izvor/korisnika. Naime, u situaciji u kojoj jedan korisnik generiše saobraćaj sa jednim tokom, a drugi korisnik generiše saobraćaj sa *n*

saobraćajnih tokova – SFQ algoritam će dati drugom korisniku *n* puta više šansi da bude opslužen u poređenju sa prvim korisnikom;

- PCQ *(Per Connection Queuing*) algoritam, predstavlja unapređenu verziju SFQ algoritma, na taj način što dvofazno koristi bafere za klasifikaciju paketa. Naime, u prvoj fazi, paketi se klasifikuju po tokovima saobraćaja (i smeštaju u primarnu FIFO grupu bafera), da bi se u drugoj fazi dodatno klasifikovali po kriterijumu pripadnosti posmatranih tokova u odnosu na izvorišnu adresu *(source-address* klasifikator), na osnovu čega se vrši pregrupisavanje paketa i njihovo smeštanje u sekundarnu grupu FIFO bafera. Nadalje, kao i u slučaju SFQ algoritma, paketi se preuzimaju – po jedan iz svakog bafera u cikličnom (Round-Robin) maniru i smeštaju u red za dodeljivanje žetona na putu ka izlazu iz paketskog filtera [11];
- RED (*Random Early Detection*) algoritam proaktivno kontroliše stepen zauzetosti paketskog reda za čekanje sa ciljem da izbegne potpunu zauzetost bafera. Kontrola stepena zauzetosti vrši se u odnosu na prosečnu dužinu reda na sledeći način: RED prati prosečnu dužinu reda i ukoliko se ova veličina počne približavati definisanom pragu, algoritam počinje nasumično da odbacuje pristigle pakete [12]. Ovakav pristup kontroli zagušenja sprečava uticaj pojave velikog stepena gubitaka paketa kod iznenadnih i velikih *burst*-ova, do kojih može u slučajevima kada se paketski red za čekanje potpuno popuni. Smanjenjem broja odbačenih paketa na kontrolisan način, RED minimizira verovatnoću pojave *tail-drop* efekta, čime se može doprineti uspostavljanju konsistentnijeg protoka, kao i manjeg kašnjenja u slučaju aktivnih tokova. Ipak, ovaj algoritam ima i nedostatak koji se ogleda u velikom broju različitih parametara, tako da je veoma komplikovan za podešavanje [13].
- FQ-CoDel (*Fair Queuing with Controlled Delay*) algoritam se smatra naprednim algoritmom koji pripada klasi algoritama za aktivno upravljanje paketskim redovima za čekanje (*Active Queue Management* - AQM). Ovaj algoritam funkcioniše tako što razdvaja saobraćaj u pojedinačne tokove i na svaki primenjuje poseban algoritam kontrolisanog kašnjenja (*Controlled Delay Active Queue Management* – CoDel, IETF RFC 8289) [14]. CoDel algoritam minimizira kašnjenje praćenjem vremena čekanja paketa i selektivnim propuštanjem paketa samo kada kašnjenja pređu određeni prag. Izolujući tokove i aktivno upravljanje kašnjenjem, FQ-CoDel algoritam obezbeđuje tzv. *pravednu* distribuciju propusnog opsega i nizak nivo kašnjenja. Posebno je efikasan u smanjenju *buffer-bloat* stanja, u kojem prekomerno baferovanje dovodi do značajnog povećanja nivoa kašnjenja u mreži [15]. FQ-CoDel algoritam je detaljno definisan u dokumentu IETF RFC 8290 [16].

4. Opis eksperimenta i dobijeni rezultati

Za potrebe uporedne analize pojedinih algoritama za upravljanje paketskim redovima za čekanje, prilikom realizacije *backup* procesa u lokalnoj računarskoj mreži, sproveden je eksperiment koji je realizovan u laboratorijskim uslovima. Eksperiment je sproveden na način koji bi trebalo da simulira realno mrežno okruženje na osnovu sledećih pretpostavki:

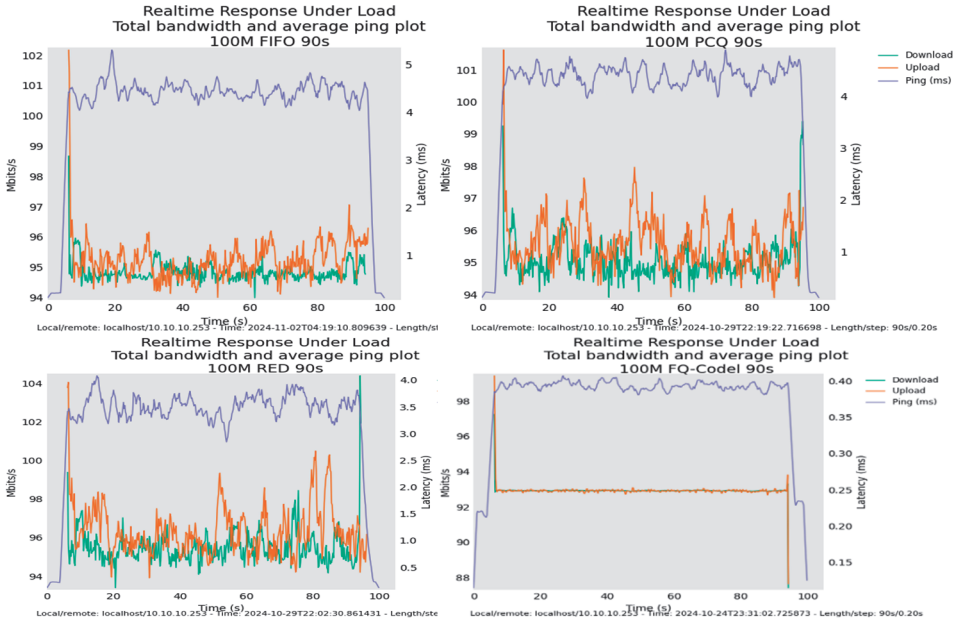
1. *backup* proces može se sprovoditi, kako van radnog vremena, tako i u toku radnog vremena. S tim da je scenario kompleksniji kada se ovaj proces sprovodi u toku radnog vremena, odnosno kada postoji veći broj mrežnih aktivnosti koji su vezani za kritične procese. Analizom svih aktivnosti došlo se do zaključka da vrednost

CIR za realizaciju *backup* procesa na posmatranoj LAN mreži može iznositi do 10% ukupno raspoloživog kapaciteta upotrebljenih mrežnih linkova;
2. svi serveri koji formiraju kopije sadržaja svojih skladišnih prostora su aktivni u toku realizacije *backup* procesa. To znači da navedeni mrežni interfejsi vrše razmenu podataka, kako za *backup* procese, tako i za sve ostale servise za koje su primarno konfigurisani;
3. svi posmatrani *host* serveri koji imaju mogućnost formiranja *snapshot*-ova, imaju mogućnost i replikacije skladišnog prostora virtuelne mašine bez njenog gašenja;
4. operativni serveri i *backup* serveri su na *Data-Link* sloju locirani u različitim VLAN-ovima. To znači da je za njihovu komunikaciju na mrežnom sloju zadužen mrežni ruter, koji sa ovim serverima razmenjuje podatke preko korespondentnih *deafult-gateway* adresa;
5. navedeni mrežni ruter je softerski orjentisan, sa operativnim sistemom koji podržava rad prikazanih paketskih redova za čekanje.

Imajući u vidu navedene pretpostavke, kao i raspoloživu opremu u trenutku sprovođenja eksperimenta, formirana je laboratorijska instalacija, koja se sastoji od upravljivog gigabitnog sviča TP-Link TL-SG3428, mrežnog rutera Mikrotik CCR200416G-2S+, dva računara sa gigabitnim portovima i sa instaliranim operativnim sistemima OpenSUSE Leap 15.6. Navedeni računari su deo hardvera neophodnog za pokretanje programskog paketa Flent (*The FLExible Network Tester*), koji je dizajniran za pokretanje i agregiranje različitih tipova mrežnih testova. Flent je posebno pogodan za merenje performansi (*benchmarking*) mreže, jer može da pruži detaljan uvid u ponašanje mreže posebno u scenarijima koji zahtevaju istovremeno testiranje različitih parametara [17]. U tom smislu, ovaj softverski paket može da oponaša opterećenje mreže u multiservisnom okruženju u realnim uslovima i da istovremeno pokrene testove kašnjenja, kao i druge *benchmarking* testove, da bi se moglo proceniti kako posmatrana mreža funkcioniše u prisustvu različitih tokova saobraćaja uključujući i scenario u kome je na mreži prisutan i mehanizam za oblikovanje saobraćaja. U tom smislu, Flent sadrži *Netperf* modul za potrebe realizacije *benchmarking*-a, za čiji rad je potrebno prisustvo dva računara koji u toku realizacije eksperimenta dobijaju uloge klijenta i servera.

Eksperiment je sproveden realizacijom *The Realtime Response Under Load* (RRUL) testa [18], koji generiše opterećenje mreže realizujući scenario „najgoreg mogućeg slučaja“ i u tim uslovima meri odziv u realnom vremenu, relativne performanse TCP i UDP tokova različitih brzina i dr. U konkretnom slučaju, upotrebljeno je klasično RRUL opterećenje mreže koje se sastoji od osam TCP tokova (po četiri u *uplink* i *downlink* smerovima), čime se pouzdano postiže zasićenje kapaciteta analiziranog linka, kao i svih bafera koji su uključeni u transport paketa [18]. RTT (*Round Trip Time*) vrednosti mere se upotrebom ICMP protokola (*ping*) i UDP protokola (*roundtrip time*). Podešeno je da se nivo ostvarenog intenziteta *backup* procesa, u prisustvu tokova drugih namena, meri generisanjem saobraćaja u trajanju 90 sec, s tim što ukupan test ima trajanje 100 sec, jer se definisanom trajanju dodaju dva perioda od po 5 sec potrebnih za formiranje tzv. *baseline* stanja na početku i na kraju testa. Na mrežnim ruteru, izvršene su sledeće pripremne radnje: definisanje VLAN-ova na *Data-Link* sloju; definisanje korespondentnih adresnih opsega i *default-gateway* vrednosti (na osnovu čega su na računarima dodeljene adekvatne fiksne IP adrese); pojedinačna podešavanja jednostrukog paketskog reda za primenu FIFO, PCQ, RED i FQ-CoDel algoritama. Ovi redovi su aktivirani - jedan po jedan i za svaki od njih je tri puta puštan u rad RRUL test. Za

uporednu analizu, uzet je najlošiji od tri dobijena rezultata testiranja, gde svaki grafikon sadrži ostvarene bitske brzine za *upload* i *download* podataka, kao i vrednost RTT dobijenu na osnovu primene *ping* testa (Slika 2).



Slika 2. Rezultati sprovedenih RRUL testova

Algoritam oblikovanja saobraćaja za sve testove podešen je na sledeći način: svi posmatrani linkovi tokom eksperimenta imaju isti kapacitet od 1 Gb/s, CIR za *korpu žetona* (TB) podešen je na 100 Mb/s (odnosno na 10% od kapaciteta posmatranih linkova), veličina *korpe žetona* definisana je na 10% CIR. Shodno navedenom, dobijeni rezultati o prenosu podataka tokom realizacije *backup* procesa u trajanju od 90 sec prikazani su u Tabeli 1.

Tabela 1. Performanse algoritama primenjenih u realizaciji backup procesa

Red	Preneto bajtova (MB)		Preneto paketa		Odbačeno paketa		CPU load (%)
	Upload	Download	Upload	Download	Upload	Download	
FIFO	1.077,6	1.089,2	1.256.176	1.252.726	26.131	24.045	10
PCQ	1.079,3	1.088,7	1.232.367	1.241.605	27.038	28.545	13
RED	1.101,5	1.079,4	1.197.670	1.203.497	24.363	25.373	12
FQ-CoDel	1.079,9	1.077,9	1.935.651	1.932.095	0	0	17

Dobijeni rezultati ukazuju da je količina podataka, koja se prenosi tokom realizacije *backup* procesa, približna u slučaju primene svih analiziranih algoritama za opsluživanje paketa u redovima za čekanje. S druge strane, rezultati sprovedenih RRUL testova (Slika 2) pokazali su da je primena FQ-CoDel algoritma u procesu oblikovanja

saobraćaja uticala na izrazito povećanje stabilnosti prenosa podataka, u poređenju sa primenom tradicionalnih algoritama. Daljom analizom Tabele 1, može se uočiti da je tokom sprovedenih testova broj prenesenih paketa u slučaju primene FQ-CoDel algoritma za oko 60% veći u odnosu na broj paketa koji je prenesen primenom drugih algoritama. To se može objasniti povećanim stepenom aktivacije TCP *slow-start* mehanizma, koji ovaj algoritam koristi tokom separacije tokova saobraćaja na veći broj redova, čime umanjuje verovatnoću pojave *burst*-ova. Ovo se može uočiti i po tome što je broj odbačenih paketa u samom filteru jednak nuli (za razliku od FIFO i PCQ algoritama, u kojima je nivo gubitaka paketa nešto manji od 2%), gde ovaj algoritam prikazuje efikasnost kontrole upravljanja baferima, u poređenju sa ostalim algoritmima (sa izuzetkom RED algoritma, kod koga je odbacivanje paketa princip rada). Treba napomenuti da se kompleksnost analiziranih algoritama tokom realizacije eksperimenata reflektovala na procesorsko opterećenje, gde su prosečne vrednosti opterećenja bile najveće u slučaju FQ-CoDel i PCQ algoritama.

5. Zaključak

Pozitivni efekti primene tehnika oblikovanja saobraćaja dodatno su izraženi izborom odgovarajućeg algoritma za upravljanje paketskim redovima, Zbog toga je u ovom istraživanju sproveden eksperiment, u kome je izvršena uporedna analiza primene ovih algoritama u slučaju kada je u LAN mreži potrebno realizovati prenose podataka inicirane serverskim *backup* procesima. Eksperiment je uključio FIFO, PCQ, RED i FQ-CoDel algoritme, nad kojima je sprovedeno testiranje upotrebom programskog paketa *Flent*. Rezultati eksperimenta pokazali su izrazitu stabilnost mrežnih tokova prilikom transporta podataka u slučaju primene FQ-CoDel algoritma. Rezultat pokazuje da primena AQM algoritama može pozitivno uticati na stabilnost tokova saobraćaja u slučajevima kada je potrebno realizovati zahtevne procese transporta podataka u prisustvu drugih mrežnih procesa koji su osetljivi na kašnjenje, odnosno na pojavu zagušenja na mreži.

Zahvalnica

Ovaj rad delimično je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije.

Literatura

[1] S. Gnanasundaram, A. Shrivastava, "Information storage and management: Storing, managing, and protecting digital information in classic, virtualized, and cloud environments", John Wiley & Sons, 2012.

[2] Cohesity, "Key concepts and best practices for snapshot backup", 17.7.2024. [Online]. Dostupno: https://www.cohesity.com/glossary/snapshot-backup/.

[3] TuxCare, "Linux KVM Backup and Recovery: Expert Tips", 24.7.2024. [Online]. Dostupno: https://tuxcare.com/blog/linux-kvm-backup-and-recovery-expert-tips/.

[4] D. Medhi, K. Ramasamy, "Traffic Conditioning", pp. 626–644, 2018.

[5] Y. Guo, "Comparison between token bucket algorithms in QoS technology", *ZTE Commun.,* Vol. 13, pp. 56–60, 2007.

[6] X. Li, "Research based on multilayer token re-allocation traffic shaping", M.S. thesis, College Comput. Sci. Technol., Nat. Univ. Defense Technol.,Changsha, China, 2010.

[7] R. Wang, W. Chi and H. Zhang, "Adaptive traffic shaping policy based on token bucket algorithm of wireless-optical broadband access network", *Journal of Electronics & Information Technology,* Vol. 39, pp. 1401–1408, 2017.

[8] H. Fu, M. Sun, B. He, J. Li and X. Zhu, "A survey of traffic shaping technology in internet of things", *IEEE Access,* Vol. 11, pp. 3794–3809, 2022.

[9] G. D. Orueta, E. S. C. Ruiz, N. O. Alonso and M. C. Gil, "Quality of service", *Industrial Communication Systems,* 2016., https://doi.org/10.1201/b16521-13.

[10] N. Harshini, *Measuring And Modeling Of Open vSwitch Performance: Implementation in Docker,* 2016.

[11] D. Iswadi, R. Adriman / R. Munadi, "Adaptive Switching PCQ-HTB Algorithms for Bandwidth Management in RouterOS", pp. 61–65, August 2019.

[12] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Transactions on Networking,* Vol. 1, pp. 397–413, 1993.

[13] T. A. Trinh and S. Molnár, "A Comprehensive Performance Analysis of Random Early Detection Mechanism", *Telecommunication Systems,* Vol. 25, pp. 9–31, January 2004.

[14] K. Nichols, V. Jacobson, A. McGregor and J. Iyengar, *IETF RFC 8289: Controlled Delay Active Queue Management,* RFC Editor, 2018.

[15] V. G. Cerf, „Bufferbloat and Other Internet Challenges," *IEEE Internet Computing,* Vol. 18, pp. 80–80, September 2014.

[16] T. Høiland-Jørgensen, P. McKenney, J. Gettys and E. Dumazet, "*The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC,* 2018.

[17] J. Kim and J. H. Lee, "Performance Evaluation of Queueing Mechanisms for Network-based Intrusion Prevention System", *SCIS & ISIS* 2010, Dec. 8-12, 2010.

[18] D. Täht, "Realtime Response Under Load (RRUL) Specification", 6.11.2012. [Online]. Dostupno: https://www.bufferbloat.net/projects/bloat/wiki/RRUL_Spec/.

Abstract: *Local network traffic management based on application of the appropriate techniques with the task of controlling the use of available network capacity in the presence of competing network activities such as backup processes. This paper will present a comparative analysis of the queue algorithms in terms of traffic shaping. The analysis includes several algorithms that are widely used in practice, including the Fair Queuing with Controlled Delay - FQ-CoDel algorithm. The aim of this comparative analysis is to examine the convenience of using the mentioned algorithms when the backup process is deployed among other services, which operate in a local area network environment.*

Keywords: *traffic shaping, backup process, packet queues, FQ-CoDel*

TRAFFIC SHAPING TECHNIQUES: APPLICATION IN BACKUP PROCESSES

Slobodan Mitrović, Valentina Radojičić