

Hardverska realizacija protokola za razmenu i ažuriranje labela u MPLS mreži

Vukota Šoškić dipl. inž, *Telekom Srbija*
dr. Aleksandra Smiljanić, *Elektrotehnički fakultet Univerziteta u Beogradu*

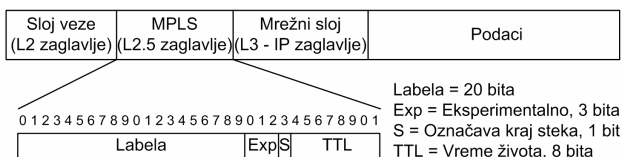
Sadržaj - Usled naglog porasta Internet mreže, povećanja saobraćaja i uvođenja novih servisa IP protokol nije bio u stanju da odgovori zahtevima u pogledu brzine obrade i prosleđivanja IP paketa. MPLS protokol je nastao da bi se obrada i prosleđivanje paketa učinili bržim i efikasnijim. LDP protokol predstavlja kontrolni protokol koji, u MPLS mreži, služi za razmenu i ažuriranje labela na bazi kojih se vrši obrada i prosleđivanje paketa. Ovaj rad se bavi arhitekturom hardverske realizacije LDP protokola koja će smanjiti angažovanost paketskog procesora i omogućiti efikasnije izvršavanje kako samog protokola, tako i drugih funkcija koje paketski procesor obavlja.

Ključne reči — FEC, LDP, labela, MPLS.

I. Uvod

Sa porastom broja uređaja na Internetu koji imaju svoje IP adrese, klasični protokoli rutiranja, koji usmeravaju pakete na bazi njegove IP adrese, postali su neefikasni, tj. nisu bili u stanju da odgovore sve većim zahtevima u pogledu brzine obrade i prosleđivanja paketa. U brzim mrežama obrada IP zaglavljaja, u realnom vremenu, može predstavljati problem, jer zahteva značajno angažovanje paketskog procesora. Kao način da se ovaj problem reši je da se prosleđivanje vrši na bazi labela (kratkih, nehijerarhijskih adresa), u okviru jedne MPLS mreže, dok se van nje prosleđivanje vrši na bazi IP adresa.

MPLS (Multiprotocol Label Switching) predstavlja standardizovani protokol, opisan u [1], koji se koristi za obradu i prosleđivanje paketa u okviru jedne mreže na osnovu labela. MPLS paketi su paketi označeni labelama, koje se dodaju u posebno zaglavljaje između L2 (link) i L3 (mrežnog) sloja (zato se protokoli prosleđivanja na bazi labela često nazivaju protokolima L2.5 sloja). Format i pozicija MPLS zaglavljaja prikazani su na Sl. 1.



Slika 1. Format i pozicija MPLS zaglavljaja

Ruteri koji su međusobno povezani i na kojima se izvršava MPLS protokol čine MPLS mrežu.

LSP (Label Switched Path) predstavlja putanju između dva rutera kroz MPLS mrežu. Ivični ruteri su ruteri koji su

povezani preko svojih interfejsa i na IP i na MPLS mrežu. Na ulazu paketa u MPLS mrežu, na ivičnim ruterima se vrši označavanje IP paketa labelom, dok se na izlazu paketa iz MPLS mreže na ivičnim ruterima vrši uklanjanje labela. Svaki ruter, na bazi svoje tabele rutiranja i dodatnih informacija vezanih za način prenosa i servise koje mreža treba da obezbedi, formira klase prosleđivanja (FEC - Forwarding Equivalence Class). FEC klasa može da uzima u obzir različite parametre: izvorišnu/odredišnu IP adresu, VPN, QoS, TE, multikast grupe... Klasa prosleđivanja određuje način na koji će paket biti prosleđen, kao i putanju (LSP) kroz MPLS mrežu. U hardverskom dizajnu koji razvijamo, sama definicija FEC klase nije od interesa, jer se u njemu realizuju distribucija i ažuriranje labela, a ne klasifikacija saobraćaja i servisa, pa ćemo pretpostaviti da je FEC klasa, npr. odredišna IP adresa mreže u koju paket treba da se prosledi. Labela pomoću koje se paket usmerava kroz MPLS mrežu predstavlja skup FEC klasa (jednu ili više njih). Prilikom ulaska u MPLS mrežu, vrši se klasifikacija paketa, da bi mu se na bazi FEC klase kojoj pripada dodelila labela. Mehanizam prosleđivanja paketa u MPLS mreži se sastoji od toga da mu se, kada MPLS paket stigne u MPLS ruter, u zavisnosti od vrednosti labela (ulazna labela) i ulaznog porta po kome je stigao, dodeljuje nova vrednost labela (izlazna labela) i da se prosleđuje na odgovarajući izlazni port. Moguće je pridružiti niz (stek) labela jednom paketu i na taj način realizovati niz servisa u MPLS mreži.

LDP (Label Distribution Protocol) je kontrolni protokol koji u MPLS mreži služi da MPLS ruteri razmene informacije na bazi kojih će formirati tabele prosleđivanja (LFIB - Label Forwarding Information Base). Na bazi labela koje služe kao direktan indeks za adresiranje LFIB tabela, omogućuje se da se prosleđivanje paketa realizuje na bazi labela koje su kraće od IP adresa. Svaki ruter podatke o tome koja labela predstavlja koji skup FEC klasa periodično šalje svojim susedima, a susedni MPLS ruteri će te labela koristiti kada mu budu prosleđivali pakete za te FEC klase. MPLS ruter će svakom od svojih suseda poslati onoliko poruka koliko ima aktivnih labela. Susedni ruteri dalje distribuiraju ove podatke svojim susedima, samo pri tom koriste različite vrednosti labela za označavanje istih FEC klasa, jer je pridruživanje labela skupovima FEC klasa lokalnog karaktera, tj. od važnosti je samo na fizičkom linku. Tako se informacije plavljenjem šire kroz MPLS mrežu i svaki MPLS ruter u mreži ima informacije o svim FEC klasama u mreži, pri čemu je potrebno sprečiti pojavljivanje petlji prilikom usmeravanja paketa pomoću labela. Ovo se može postići na taj način što se, pored toga što svaka poruka nosi informacije o tome na koji se skup FEC klasa odnosi konkretna labela, pridodaje ID rutera koji je taj skup FEC klasa oglosio. Kada MPLS ruter primi LDP poruku po

interfejsu preko koga on inače ne bi slao, klasičnim protokolom za rutiranje, pakete ka ruteru čiji se ID nalazi u LDP poruci, on tu poruku ignoriše, jer bi prihvatanjem te poruke i njenim daljim prosleđivanjem svojim susedima, napravio petlju. Posledica ovoga bi bilo dodeljivanje više labela istom skupu FEC klasa i neoptimalno rutiranje saobraćaja putem MPLS protokola. Viši protokoli (koji rade sa IP adresama i koriste tabelu rutiranja), svakom interfejsu prosleđuju informacije o tome koji je ruter dostupan preko tog interfejsa na bazi njegovog ID-a. Dakle, MPLS ruter dobija potvrdu od suseda kada ovaj primi LDP poruku o ažuriranju labela, ali ne zna da li je to njegov sused uvrstio u LFIB tabelu i nastavlja da mu periodično šalje LDP poruke. Kako neke od ovih poruka sused odbacuje, one čine nekoristan saobraćaj, ali ukoliko bi neki link otkazao, ove poruke pomažu da se LSP putanje rekonfiguriraju i mreža oporavi. Naime, ako je preko tog interfejsa ruter čiji se ID nalazi u poruci, postao dostupan na bazi tabele rutiranja, sused će početi da prihvata LDP poruke i da pomoću njih ažurira svoju LFIB tabelu. Da bi se sprečilo neefikasno rutiranje u slučaju da dva ivična rutera, van MPLS mreže, mogu da dosegnu istu destinacionu mrežu, u poruci koja sadrži mapiranje labela - FEC, ivični ruter može preneti i podatak sa kojom metrikom koju mrežu može da dosegne. Bitno je uočiti da MPLS ruter ne donosi odluku o klasifikaciji paketa na bazi sopstvene tabele rutiranja, nego na bazi podataka koje dobija od susednih rutera, a oni vode poreklo od ivičnih rutera koji oglašavaju informacije o tome koje mreže mogu da dosegnu preko svojih IP interfejsa. Detalji LDP protokola su dati u [2]-[4]. Takođe je bitno uočiti da se LFIB tabela koristi za prosleđivanje korisnog saobraćaja u jednom smeru, a tabela koju koristi LDP protokol služi za prosleđivanje kontrolnih paketa u suprotnom smeru.

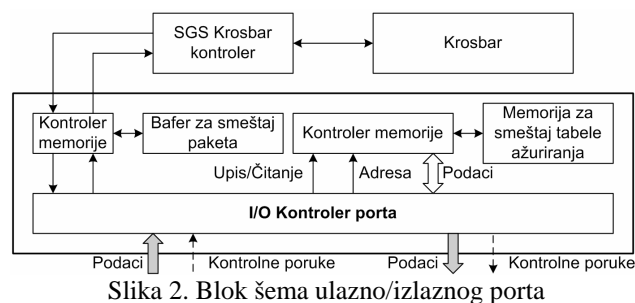
Budući da se razmena LDP poruka odvija između dva susedna MPLS rutera, može se koristiti LLC (Logic Link Control) protokol koji treba da obezbedi pouzdan prenos podataka po linku, a on bi se, takođe, realizovao u hardveru. LLC protokol obavlja funkciju uspostavljanja, održavanja i raskidanja logičke veze između dva LLC čvora. Samo zaglavlje LLC protokola je smešteno u link sloj protokolskog steka. Detalji protokola su opisani u [5].

II. Hardverska realizacija LDP protokola

Osnovne funkcije hardverskog dizajna koji će biti opisan u ovom radu su prosleđivanje i ažuriranje labela u okviru jedne MPLS mreže. Prosleđivanje labela obuhvata: inicijalno slanje labela i skupa FEC klasa koje ona predstavlja, slanje periodičnih poruka koje signaliziraju da je labela još aktivna i da nije došlo do promena u skupu FEC klasa koje ta labela predstavlja, slanje poruka koje nose informacije o promenama u skupu FEC klasa koje ta labela predstavlja, a kao proširenje ovog protokola, može se uvesti i slanje poruka koje ukidaju važenje trenutno validnih labela. Na osnovu primljenih poruka, na portu rutera se ažurira LFIB tabela, a ažuriraju se i pomoćne tabele koje služe za njeno formiranje i održavanje. Ažuriranje labela je periodično, a labela koja nije ažurirana, za vreme koje je nešto duže od trostrukog vremena predviđenog za ažuriranje, briše se iz LFIB tabele. Ovo za posledicu ima da taj MPLS ruter neće slati susedima periodične poruke za održavanje te labela, pa će

one redom zastarevati i biti brisane iz LFIB tabele svih MPLS rutera. Podaci ulazna labela i ulazni port predstavljaju indeks LFIB tabele, a podaci koji taj indeks adresira su izlazna labela i izlazni port. Praktično kada stigne paket, na dati ulazni port, koji nosi koristan sadržaj i označen je nekom (ulaznom) labelom, na osnovu ulaznog porta i labela kontroler čita iz memorije gde taj paket treba proslediti i koju mu labelu treba pridružiti. Informacije o labelama se šalju u smeru suprotnom od toka podataka kroz MPLS mrežu, tj. da se figurativno izrazimo, uzvodno (upstream). To znači da, prilikom ažuriranja labela, ruter šalje uzvodnom susedu podatak o labeli kojom treba da mu se označi neki skup FEC klasa. Za susedni ruter koji je tu labelu primio, ta labela je izlazna labela i njoj pridružuje jednu svoju slobodnu labelu, koju šalje svom uzvodnom susedu da bi ga obavestio o tome kako da mu označi saobraćaj za prosleđeni skup FEC klasa i tu labelu smatra svojom ulaznom labelom. Mehanizam ažuriranja labela se vrši na taj način što na ulazni deo I/O porta stigne izlazna labela kojoj treba ili dodeliti ulaznu labelu (ako već nije) ili ažurirati već dodeljenu. Važno je napomenuti da se labela koriste kao direktne adrese (ili se jednostavno mogu, linearnom transformacijom, jednoznačno, pretvoriti u adrese) kojim se adresiraju tabele neophodne za njihovo ažuriranje.

Bez umanjjenja opštosti opis hardverske implementacije biće dat na primeru implementacije jednog ulazno/izlaznog porta na FPGA čipu, čija je blok šema prikazana na Sl. 2.



Slika 2. Blok šema ulazno/izlaznog porta

U I/O kontroleru porta se izvršava LDP protokol. Memorija za smeštaj tabele ažuriranja sadrži podatke o labelama, njihovom statusu, kao i druge informacije neophodne za funkcionisanje protokola, dok se preko kontrolera te memorije vrši komunikacija sa I/O kontrolerom porta. U ovom radu se sve operacije sa labelama vrše putem tabele ažuriranja. Bafer za smeštaj paketa je memorija u koju se smeštaju paketi podataka koji čekaju u redu da budu preneti, preko krosbara, ostalim portovima. Kontroler memorije upravlja baferom i inicira prenos paketa preko krosbara. Krosbar kontroler upravlja prosleđivanjem paketa preko krosbara.

Tabelu ažuriranja koristi LDP protokol, a na bazi nje se formira LFIB tabela koja služi, MPLS protokolu, za prosleđivanje paketa koji nose koristan saobraćaj. Kako se ovaj rad bavi samo realizacijom LDP protokola, a ne i realizacijom MPLS protokola, LFIB tabela nije prikazana.

Hardverski dizajn je opisan: formatom poruka koje susedni MPLS ruteri razmenjuju, protokolom koji će definisati način obrade podataka, strukturom podataka koje hardver treba da obradi i prikazom arhitekture hardvera koji ove funkcije treba da obavi.

A. Format poruka koje susedni MPLS ruteri razmenjuju

Što se tiče formata poruka koje susedni MPLS ruteri razmenjuju bitno je da se napomene da te poruke moraju da sadrže labelu, tip poruke (*Initial*, *Update*, *Change*, *Invalid*), spisak FEC klasa i ID rutera koji je te klase oglasio. Za poruku tipa *Initial* spisak sadrži klase koje data labela predstavlja, *Update* poruka šalje vrednost labele koju treba ažurirati, poruka tipa *Change* sadrži spisak FEC klasa koje treba dodati/ukloniti, dok poruka tipa *Invalid* šalje vrednost labele koju treba ukloniti iz tabele prosleđivanja ne čekajući da ona zastari (poruku *Invalid*, u ovoj implementaciji, ne inicira sam LDP protokol, ali je predviđena zbog proširenja protokola).

B. Protokol za ažuriranje labele

Protokol preko koga se obavlja ažuriranje labele sastoji se od dva dela:

- Protokol za ažuriranje slobodnih labele u tabeli ažuriranja, koji je periodično pretražuje i proverava rokove važenja labele i labele koje nisu validne stavlja u red slobodnih labele odakle se one, kasnije, mogu dodeljivati.
- Protokol za procesiranje LDP poruka koji, po prijemu LDP poruka, ažurira validnost odgovarajućih labele i obaveštava susede o tome. Poruke o ukidanju labele nije potrebno slati, mada protokol predviđa njihovu obradu.

V -bit označava validnost labele, S -bit označava da li je slobodna labela već stavljena u red, T_s -vreme koje pokazuje sat realnog vremena i T_u -trenutak upisa labele u memoriju. Oznaka $X(i)$ označava podatak/polje na adresi i .

Protokol za ažuriranje slobodnih labele u tabeli ažuriranja se sastoji iz sledećih koraka:

- Inicijalno setovati adresu $i=1$, $i=[IzlazniPortIzlaznaLabela]$.
- Da li je labela na adresi i validna ($UlaznaLabela(i)$), tj. da li je $V(i)=1$? Ako nije preći na korak 5.
- Da li je $T_s-T_u(i)<T_{max}$? Ako jeste preći na korak 7.
- Setovati $V(i)=0$. Preći na korak 6.
- Da li je $S(i)=1$? Ako jeste preći na korak 7.
- Staviti labelu $UlaznaLabela(i)$ u red slobodnih labele. Setovati $S(i)=1$.
- Da li je $i=i_{max}$? Ako jeste preći na korak 1.
- Inkrementirati i , tj. $i=i+1$. Preći na korak 2.

Protokol za procesiranje LDP poruka se, u ulaznom delu I/O kontrolera porta, sastoji iz sledećih koraka:

- Čekati dok ne stigne, na ulazni port, LDP poruka iz mreže, pa onda preći na korak 2.
- Da li je poruka stigla preko odgovarajućeg porta (proverava se *RuterID*)? Ako nije prelazi se na korak 1.
- Preko krosbara se, svim portovima šalje ista poruka koja sadrži tip poruke, izlazni port, izlaznu labelu i *RuterID*. Preći na korak 1.

Protokol za procesiranje LDP podataka se, u izlaznom delu I/O kontrolera porta, sastoji iz sledećih koraka:

- Čekati dok ne stigne, preko krosbara, LDP poruka sa nekog od portova, pa onda preći na korak 2.
- Da li je poruka *Invalid*? Ako jeste ide se na korak 8.
- Da li je poruka *Initial*? Ako nije ide se na korak 7.
- Da li je red slobodnih labele prazan? Ako nije prelazi se na korak 6.
- Zahtev za dodelu slobodnih labele se odbija i šalje se poruka o grešci. Prelazi se na korak 1.

6. Adresira se memorija za smeštaj tabele ažuriranja na bazi podatka iz polja *IzlazniPort* i *IzlaznaLabela*. Na toj lokaciji se upisuje $V=1$ i $S=0$, u polje *UlaznaLabela* se upisuje prva slobodna labela (mekanizam evidentiranja i dodele slobodnih labele je sledeći: Inicijalno su sve labele slobodne i sve se stavljaju u red slobodnih labele (upisuju se redom u deo memorije predviđen za to). Dva pokazivača ukazuju na početak i kraj reda (prva i poslednja nedodeljena labela). Kako se labele dodeljuju, pročita se vrednost labele sa lokacije početka i ta labela se dodeli, pa se pokazivač početka inkrementira, a kako se labele oslobađaju, pokazivač kraja reda se inkrementira i na tu lokaciju se upiše labela čija je važnost istekla. Ovo je ciklično pomeranje u jednom memorijskom bloku i kada bilo koji od ova dva pokazivača stigne do kraja reda on se nastavlja na njegovom početku. Takođe se mora voditi računa o tome da li je pokazivač početka reda na nižoj ili višoj lokaciji od pokazivača kraja reda, da bi se ispravno protumačilo u kom opsegu se nalaze slobodne labele), a u polje T_u se upisuje vreme koje pokazuje sat realnog vremena. Na izlaz porta se šalje poruka koja sadrži vrednost *UlaznaLabela* i FEC koje joj pripadaju. Zatim se prelazi na korak 1.

7. Adresira se memorija za smeštaj tabele ažuriranja na bazi podatka iz polja *IzlazniPort* i *IzlaznaLabela*. U polje T_u se upisuje vreme koje pokazuje sat realnog vremena. Na izlaz porta se šalje poruka koja sadrži vrednost *UlaznaLabela* i spisak FEC koje treba, eventualno, dodati/ukinuti. Zatim se prelazi na korak 1.

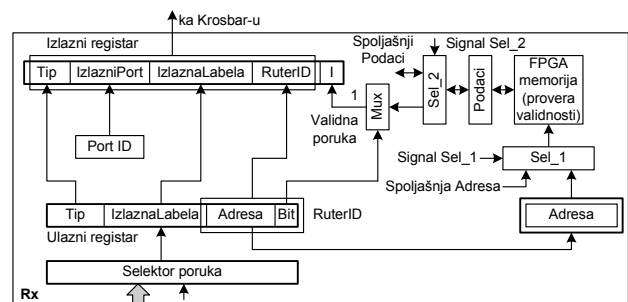
8. Adresira se memorija za smeštaj tabele ažuriranja na bazi podatka iz polja *IzlazniPort* i *IzlaznaLabela*. Na toj lokaciji se upisuje $V=0$. Ova labela će moći da se ponovo koristi kada protokol koji proverava validnost labele pročita ovaj zapis (detektuje $V=0$) i labelu iz polja *UlaznaLabela* stavi u red slobodnih labele, uz postavljanje $S=1$. Preći na korak 1.

Protokolu se mogu dodati akcije i poruke o greškama u slučaju prijema neregularnih poruka koje je susedni MPLS ruter poslao (u smislu sadržaja ili redosleda pristizanja).

Bitno je napomenuti i da konvergencija LDP protokola direktno zavisi od brzine konvergencije samog protokola rutiranja, kao i od brzine razmene i obrade LDP poruka.

C. Arhitektura hardvera u kome se obavlja ažuriranje i prosleđivanje labele

Na Sl. 3 prikazan je ulazni deo I/O porta.

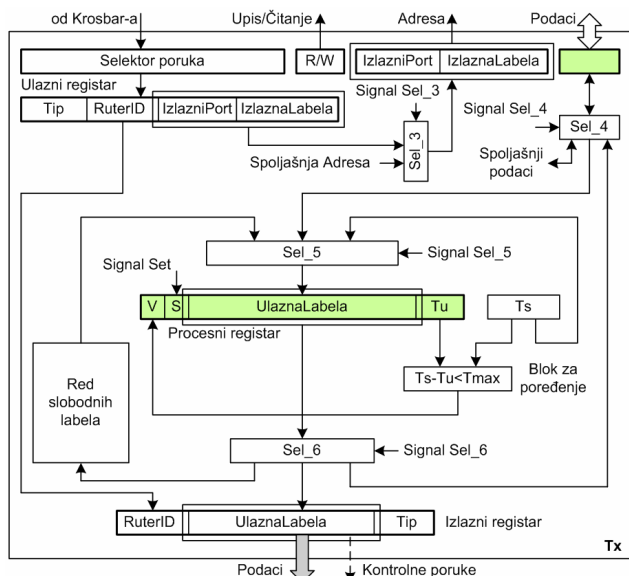


Slika 3. Ulazni deo I/O kontrolera porta

Selektor poruka izdvaja odgovarajuće LDP poruke i u ulazni registar smešta informacije o tipu poruke, izlaznoj

labeli i *RuterID*-u. Polje *RuterID* adresira konkretan bit (*I*) da bi se utvrdilo da li se dati ruter (predstavljen preko svog *RuterID*-a) može dosegnuti preko ovog interfejsa, a realizuje se tako da se *RuterID* deli na dva dela od kojih jedan, *Adresa*, adresira lokaciju u memoriji preko adresne linije, a drugi, *Bit*, koji bira konkretan bit adresiranog podatka preko multipleksera. Na taj način se štedi memorijski prostor, jer jedan zapis u memoriji sadrži podatke za više rutera. Podatke u ovu memoriju upisuju spolja viši protokoli preko signala *Spoljašnja adresa*. Tako se na bazi vrednosti bita *I*, odlučuje da li se preko krosbara prosleđuje poruka iz izlaznog registra ili ne. Ukoliko je $I=1$ ova poruka se difuziono šalje svim ostalim portovima.

Na Sl. 4. prikazan je izlazni deo I/O kontrolera porta.



Slika 4. Izlazni deo I/O kontrolera porta

Selektor poruka izdvaja odgovarajuće LDP poruke i podatke, o tipu poruke, izlaznom portu po kome je stigla ta poruka i izlaznoj labeli, prosleđuje ulaznom registru. Ovaj registar služi da se preko njega adresira deo memorije u kome je smeštena tabela za ažuriranje, a u tip poruke koja je pristigla sa krosbara određuje koje će operacije I/O kontroler izvršiti. Adresni, R/W i registar podataka služe za komunikaciju sa memorijom u kojoj je smeštena tabela za ažuriranje labela. U registru T_s je sat realnog vremena, vrednosti T_s i T_u se porede u bloku za poređenje koji treba da, kao izlaz, da podatak o tome da li je isteklo vreme važenja labela, a red slobodnih labela sadrži labela koje se trenutno ne koriste. U Procesnom registru se vrše sve operacije nad podacima smeštenim u tabelu ažuriranja u skladu s opisanim protokolom. Izlazni registar sadrži podatke o tipu poruke, ulaznoj labeli i *RuterID* rutera sa koga ova poruka potiče i koje treba poslati susednom ruteru.

Blok *Sel_5* vrši selekciju podataka koji se upisuju u Procesni registar, a u zavisnosti od toga da li je reč o dodeljivanju labela iz reda slobodnih labela, ažuriranju labela ili njenom ukidanju. Blok *Sel_6* vrši selekciju jedne od operacija: upisivanje slobodnih labela u red slobodnih labela, prosleđivanje labela susednom ruteru ili upis u memoriju vremena koje pokazuje sat realnog vremena. Viši protokoli koriste linije *Spoljašnja adresa* i *Spoljašnji podaci* kako bi mogli da upisuju podatke u delove

memorije I/O porta neophodne za funkcionisanje LDP protokola. Ako u radu protokola a_1 i a_2 dođe do konflikta u pokušaju da se pristupi resursima memorije ili registrima, prioritet u izvršavanju operacija ima protokol koji obrađuje pristigle poruke od susednih MPLS rutera.

Dodeljivanje labela je po portu, tj. svaki port sam odlučuje o tome koje će labela dodeljivati kada formira vezu *IzlaznaLabela-UlaznaLabela*. Treba napomenuti da se mora realizovati, u okviru svakog I/O porta, u samom hardveru (na FPGA čipu), tabela koja će se koristiti za prosleđivanje korisnog saobraćaja (LFIB), tj. *UlaznaLabela* će biti adresa koja će adresirati lokaciju na kojoj će biti podatak na koji će *IzlazniPort* paket da se prosledi i koja *IzlaznaLabela* će mu biti pridružena, ali ovo je deo MPLS protokola (protokol prosleđivanja podataka), a ne LDP protokola (kontrolni protokol).

Ovim je završen opis hardvera koji treba da obavlja funkciju ažuriranja i prosleđivanja labela. Detaljan opis hardvera i softverskih alata se može naći na sajtu [6].

III. Zaključak

Prednosti ovakve realizacije LDP protokola su u tome što se ažuriranje i prosleđivanje labela ubrzava i smanjuje angažovanost paketskog procesora na obradi LDP poruka. Ovakva realizacija protokola smanjuje mrežni saobraćaj prilikom razmene labela, a koji u slučaju velikog broja aktivnih labela može biti značajan. Takođe, ovako realizovan protokol razmene, omogućava da se MPLS mreža rekonfiguriše u slučaju otkaza nekog linka. Ovaj protokol, uz relativno jednostavna proširenja, može i da podrži i razne servise koje savremena MPLS mreža mora da obezbedi (VPNs, Tunneling, TE, ...). Jednostavno je obezbediti, a u saradnji sa višim protokolima na ivičnim ruterima, da se inicira slanje labela u on-demand modu. Tako bi se moglo obezbediti da se putanje kroz MPLS mrežu uspostavlja samo kada postoje paketi za te FEC klase (destinacione mreže). Uz izvesne modifikacije, ovaj protokol može da prenosi i multikast saobraćaj.

Literatura

- [1] RFC 3031: Multiprotocol Label Switching Architecture
- [2] RFC 5036: LDP Specification
- [3] Thomas D. Nadeau, "MPLS Network Management MIBs, Tools, and Techniques"
- [4] Adrian Farrel, "The Internet and Its Protocols - A Comparative Approach"
- [5] Part 2: Logical Link Control, Adopted by the ISO/IEC and redesignated as ISO/IEC 8802-2:1998
- [6] www.xilinx.com

Hardware realization of exchanging and updating labels protocol in MPLS network

Šoškić Vukota dipl.inž, dr. Aleksandra Smiljanić

Abstract - As the Internet is growing and its new services are developing, the IP protocol will not be able to fulfill the demands on the processing and forwarding speeds. MPLS was proposed in order to make the packet forwarding faster and more efficient using labels which are shorter than IP addresses. LDP protocol is the control protocol for the exchange and update of the information about active labels. In this paper, we will present the architecture of the hardware realization of the LDP protocol which will decrease the processing burden of the packet processor and provide more efficient execution of the LDP protocol and of other protocols that share the resources.