

Razvoj upravljačke komponente za ostvarenje zahtjeva usluga unutar mreže pružatelja internetskih usluga

Šlezak, Jakov

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:567135>

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

Download date / Datum preuzimanja: **2025-03-22**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 365

**RAZVOJ UPRAVLJAČKE KOMPONENTE ZA OSTVARENJE
ZAHTJEVA USLUGA UNUTAR MREŽE PRUŽATELJA
INTERNETSKIH USLUGA**

Jakov Šlezak

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 365

**RAZVOJ UPRAVLJAČKE KOMPONENTE ZA OSTVARENJE
ZAHTJEVA USLUGA UNUTAR MREŽE PRUŽATELJA
INTERNETSKIH USLUGA**

Jakov Šlezak

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 365

Pristupnik: **Jakov Šlezak (0036524713)**

Studij: Računarstvo

Profil: Znanost o mrežama

Mentor: prof. dr. sc. Željko Ilić

Zadatak: **Razvoj upravljačke komponente za ostvarenje zahtjeva usluga unutar mreže
pružatelja internetskih usluga**

Opis zadatka:

U svijetu sve većih zahtjeva za mrežnim resursima, optimizacija mrežnog prometa postaje ključna za osiguranje kvalitetne usluge. Implementacijom raznih protokola i funkcionalnosti unutar mreže ti se zahtjevi mogu lakše i učinkovitije ispuniti. Vaš zadatak je osmisliti i realizirati upravljačku komponentu unutar mreže koja kao ulaz prima određene zahtjeve za neku aplikaciju i/ili uslugu, te osigurava ispunjenje primljenih zahtjeva unutar mreže pružatelja internetskih usluga. Potrebno je identificirati i analizirati na koji način mrežni elementi unutar mreže pružatelja internetskih usluga moraju biti konfigurirani za tehničko ostvarenje takve upravljačke komponente. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 28. lipnja 2024.

Zahvala

Zahvaljujem kolegama iz tvrtke Combis što su mi dali na raspolaganje datoteke virtualnih usmjeritelja za simulaciju.

Zahvaljujem kolegama iz tvrtke Omonia na računalnim resursima za potrebe simulacije.

Zahvaljujem svojoj široj i užoj obitelji na pomoći i podršci svih ovih godina studiranja.

Na kraju posebno želim zahvaliti prof. dr. sc. Željku Iliću na izvrsnom mentorstvu, vodstvu, dobrim savjetima i strpljenju tijekom cijelog diplomskog studija.

Sadržaj

Uvod	1
1. MPLS i tradicionalni TE	2
1.1. Osnovni koncepti MPLS-a	2
1.2. MPLS-TE	4
2. Segment routing.....	6
2.1. Osnove Segment routinga.....	6
2.2. SR politike i identifikacija SR politika.....	8
2.3. SR putanje	8
2.3.1. Eksplicitne putanje	9
2.3.2. Dinamične putanje.....	12
2.4. PCE.....	19
3. Simulacija i konfiguracija.....	21
3.1. Simulator i virtualizacija	21
3.2. Konfiguracija usmjerenja	22
3.2.1. Konfiguracija sučelja i ISIS-a.....	22
3.2.2. Konfiguracija SR-TE-a.....	25
4. Upravljačka skripta.....	28
4.1. Osnovna struktura i preduvjeti	28
4.2. Kreiranje i brisanje Xconnect P2P servisa	29
4.3. Dohvaćanje topologije, računanje i primjenjivanje izračunatih TE metrika	33
4.4. Izračun margina za SR politike s aktivnom dinamičnom putanjom s TE metrikom	37
4.5. Automatski izračuni TE metrike i margine	42
4.6. Nedostaci skripte u računanju TE metrika i margina	43
Zaključak	45

Literatura	46
Sažetak.....	49
Summary.....	50
Privitak	51

Uvod

U današnje digitalno doba, gdje se razna poduzeća (i pojedinci) sve više oslanjaju na internetske usluge, osiguravanje zadovoljavajućih performansi i pouzdanosti postaje sve teže za pružatelje internetskih usluga (ISP-ove), a kako se mrežni promet povećava i postaje sve složeniji, upravljanje mrežnim resursima i ispunjavanje različitih zahtjeva za uslugama predstavlja značajan izazov.

Iako je osnovni *Multi Protocol Label Switching* (MPLS) često korišten za potrebe *Traffic Engineeringa* (TE), povećavanjem korisnika i mrežnog prometa pokazao se kao problematičan u području skalabilnosti. Modernija arhitektura pod nazivom *Segment Routing* (SR) rješava problem skalabilnosti MPLS-a unutar TE-a i ne zahtijeva posebne signalne protokole poput LDP-a i RSVP-TE-a. SR je interoperabilan sa MPLS-om i time omogućava ISP-u jednostavni tranzicijski period bez potrebe velikih pojedinačnih ulaganja, a ne zahtijeva nikakve dodatne protokole, već distribuira informacije putem postojećih *Interior Gateway Protocola* (IGP) poput ISIS-a ili OSPF-a, a može koristiti i BGP-LS.

Fokus ovog rada je na pojednostavljenju i optimizaciji konfiguracije mrežnih uređaja primjenom SR-a i TE-a, posebno u kontekstu pružanja L2VPN P2P usluga. Za te potrebe razvijena je python skripta koja omogućuje jednostavno kreiranje L2VPN P2P usluga, koje se oslanjaju na kreirane SR politike s određenim parametrima te novom inovativnom načinu izračuna i optimizacije metrika i margina koje te SR politike koriste kako bi se zadovoljili traženi zahtjevi.

Kako bi se osiguralo realistično okruženje za testiranje i razvoj, korišten je GNS3 simulator u kojem je virtualizirano više Cisco IOS XRv9000 usmjerivača koji su međusobno povezani u topologiju koja simulira mrežu (ili dio mreže) ISP-a.

1. MPLS i tradicionalni TE

U ovom poglavlju rad se osvrće na osnovne koncepte MPLS-a i njegovu primjenu u mreži za TE na tradicionalan način. Rad se dotiče i uspoređuje signalne protokole koji se koriste unutar mreža s klasičnim MPLS i TE primjenama .

1.1. Osnovni koncepti MPLS-a

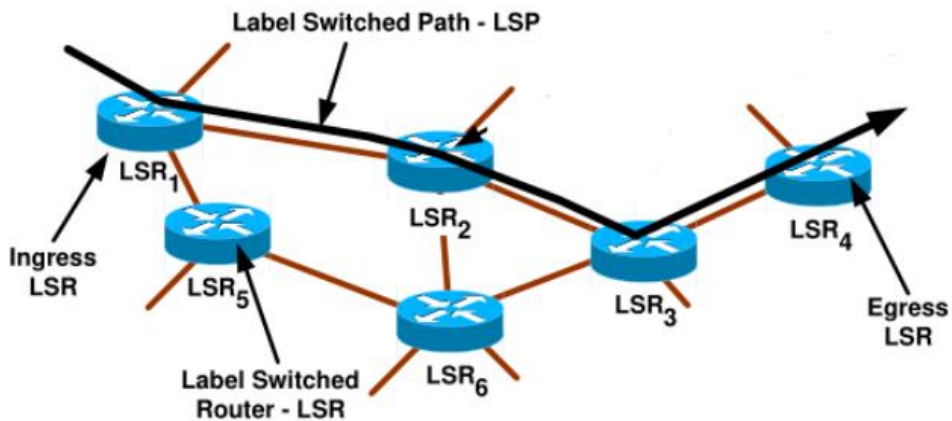
MPLS je tehnologija unutar telekomunikacijskih mreža koja, za razliku od tradicionalnog IP usmjeravanja koje se oslanja na IP adrese i tablice usmjeravanja, omogućava usmjeravanje prometa pomoću labela[1]. Ova tehnologija omogućava kreiranje virtualnih puteva između krajnjih točaka koje su poznate kao *Label Switched Paths* (LSP-ovi). LSP-ovi su unaprijed definirane putanje koje se koriste za prosljeđivanje paketa, što omogućava optimizaciju prosljeđivanja mrežnog prometa i smanjenje latencije jer nema potrebe za procesorski kompliciranim pretraživanjem tablica usmjeravanja.

Forwarding Equivalence Class (FEC) je skup nekih paketa koji se obrađuju na isti način, tj. slijede istu putanju kroz MPLS mrežu. Ti razni paketi se grupiraju u FEC-ove na temelju određenih kriterija poput npr. odredišne adrese ili *Quality of Service* (QoS) zahtjeva. Kada paket bude dodijeljen određenom FEC-u, on će putovati kroz MPLS mrežu po unaprijed definiranom LSP-u, tj. vjerojatno će im se dodijeliti ista labela.

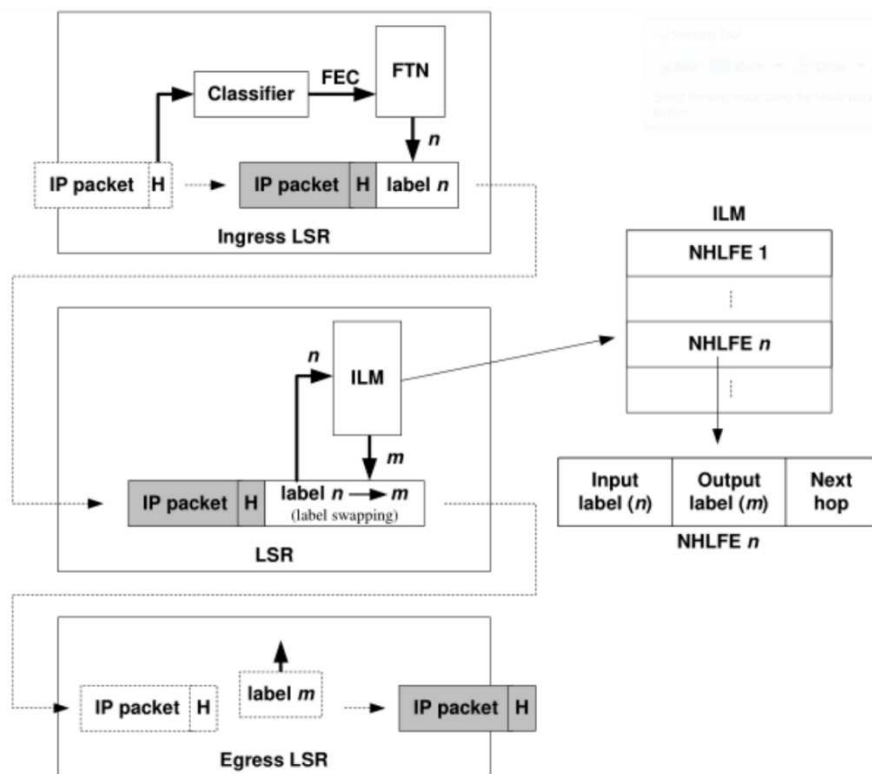
Label Switch Routeri (LSR-i) su usmjerivači koji koriste labele za usmjeravanje prometa kroz mrežu. LSR-i imaju *Incoming Label Map* (ILM) tablicu u kojoj se nalazi više *Next Hop Label Forwarding Entrya* (NHLFE-a) koji opisuju vrste operacija koju LSR treba obaviti za pojedini paket s nekom labelom. LSR-i općenito obavljaju tri (glavne) funkcije [1][3]:

1. *Label Push*: Prilikom ulaska nekog paketa u MPLS mrežu, ulazni LSR (nazvan još i ulazni *Label Edge Router* – LER) dodaje labelu na paket preko FEC to NHLFE (FTN) mape i time definira NHLFE za tu vrstu paketa.
2. *Label Swap*: Unutar MPLS mreže, svaki sljedeći LSR mijenja labelu paketa s novom labelom, te paket s novom labelom šalje dalje na sučelje definirano u NHLFE-u
3. *Label Pop*: Na izlazu iz MPLS mreže, izlazni LER uklanja labelu i prosljeđuje paket dalje prema odredištu pomoću klasičnog usmjeravanja.

Kako bi LSR-ovi znali koju labelu primijeniti na paket prilikom *Label Push* ili *Label Swap* operacije, najčešće se koristi *Label Distribution Protocol* (LDP). LDP je protokol koji se koristi za razmjenu labela između LSR-ova, te omogućava da LSR-ovi uspostave i održavaju LSP-ove kroz mrežu. Skice MPLS mreže i funkcije LSR-a mogu se vidjeti na Slika 1.1 i Slika 1.2.



Slika 1.1 Skica MPLS mreže, Izvor: preuzeto iz [2]



Slika 1.2 Skica funkcija LSR-a, Izvor: preuzeto iz [2]

1.2. MPLS-TE

MPLS je već dugo vremena temeljna tehnologija koja se koristi za TE u IP mrežama jer omogućuje eksplicitno usmjeravanje prometnih tokova u obliku LSP-a. No u MPLS-TE-u, LSP-ovi se kreiraju i održavaju uzimajući u obzir zahtjeve za mrežnim prometom i resursima. Jedan od ključnih protokola za implementaciju MPLS-TE-a je *Resource Reservation Protocol – Traffic Engineering* (RSVP-TE). Taj protokol omogućava kreiranje i održavanje LSP-ova s posebnim zahtjevima za mrežni promet i resurse. RSVP-TE je zapravo proširenje osnovnog RSVP-a kako bi omogućio rezervaciju resursa duž puta kojim bi će neki promet teći i osigurao da svaki link na putu ima dovoljno kapaciteta za zadovoljenje zahtjeva LSP-a.

Proces uspostavljanja LSP-a pomoću RSVP-TE-a uključuje [4]:

- *PATH* poruku: Inicijator (najčešće LER) šalje *PATH* poruku duž željene putanje prema konačnom odredištu LSP-a. Ova poruka sadrži zahtjeve za resursima.
- *RESV* poruku: Svaki LSR na putu LSP-a rezervira resurse prema opcijama specificiranim u *PATH* poruci te ju prosljeđuje dalje. Kada *PATH* poruke prođu sve LSR-ove do izlaznog LER-a, on vraća po istoj putanji *RESV* poruku u kojoj se nalazi vrijednost rezervirane labele kako bi prethodni LSR znao koju labelu staviti na paket prilikom prosljeđivanja paketa i *SWAP* operacije. Tako svi LSR-ovi vraćaju unatrag *RESV* poruku dok ona ne dođe do inicijatora, čime se LSP smatra uspostavljenim.

Glavne razlike između LDP-a i RSVP-TE-a se mogu vidjeti u Tablica 1.1. LDP se često koristi za jednostavno dinamičko mapiranje IP ruta na labele, nema općenito mogućnosti TE-a, *Fast Reroutinga* (FRR) i mogućnosti rezervacije *Bandwidtha* (BW) kao RSVP-TE, ali zato podržava *Equal-Cost Multi-Path routing* (ECMP) i skalabilniji je.

Tablica 1.1 Razlike LDP-a i RSVP-TE-a, Izvor: preuzeto iz [5]

	LDP	RSVP-TE
Jednostavnost konfiguracije, korištenja i otklanjanje poteškoća	Jednostavno	Teško
FRR	Ne	Da
ECMP	Da	Ne
TE	Ne	Da

Rezervacija BW-a	Ne	Da
Način transporta poruka	TCP	IP (UDP)
Skalabilnost	Visoka	Osrednja
LSP topologija	MPtP	PTP

2. Segment routing

Segment Routing (SR) je modernija mrežna tehnologija razvijena iz potrebe za pojednostavljenjem konfiguracije i kao odgovor na ograničenja i kompleksnost tradicionalnih MPLS-TE sustava. Razvoj SR-a predvodio je Cisco [6] koji ga je prvi implementirao još tijekom 2013. godine. U sljedećim potpoglavljima rad opisuje osnove rada SR-a uključujući definiranje SR politika i njihovih putanja sa različitim vrstama ograničenja za ostvarenje TE-a.

2.1. Osnove Segment routinga

SR koristi koncept usmjeravanja gdje izvorište (ili početak SR mreže) određuje put kojim će paket proći kroz mrežu, a ostatak mreže samo izvršava naredbe koje su dodane u paket i koje se prenose zajedno s paketom. Segment označava identifikator za neku instrukciju, npr. prosljeđivanje prema nekom odredištu. SR iskorištava MPLS podatkovnu ravninu bez dodatnih izmjena, a MPLS labele koje identificiraju segmente nazivaju se Segment Identifikatori (engl. *Segment ID*, skraćeno SID)

Postoje dvije vrste segmenata:

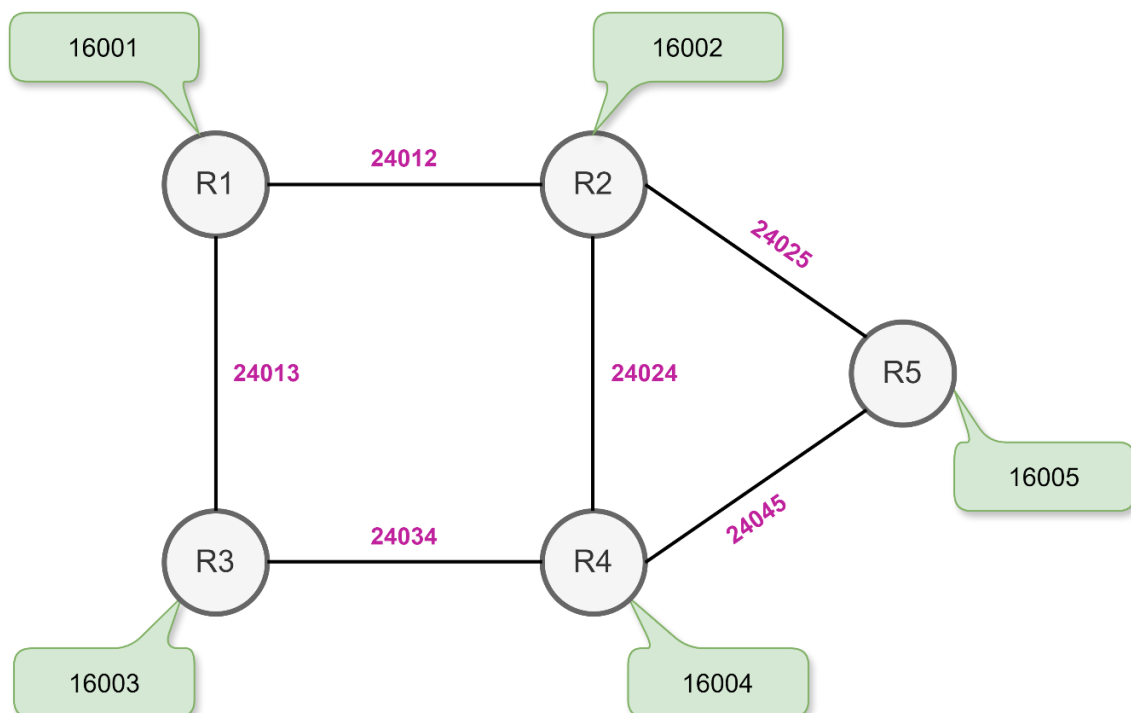
- Globalni segmenti (engl. *Global segments*): To su segmenti koji su poznati svakom čvoru unutar SR domene. Svaki čvor dodaje instrukciju koju segment predstavlja u svoju tablicu prosljeđivanja. Unutar MPLS-a, za te segmente koristi se skup *Segment Routing Global Block* (SRGB) labela. Tvornički zadane vrijednosti SRGB-a su vrijednosti od 16.000 do 23999, te se najčešće i koriste u produkciji. Globalni segment se najčešće opisuje kao početni broj SRGB-a (npr. 16000) zbrojen s vrijednosti indeksa koji mora biti jedinstven unutar SR domene. [7]
- Lokalni segmenti (engl. *Local Segments*): Samo čvor koji je definirao segment zna za njega i instrukciju koju predstavlja. Unutar MPLS-a to je lokalno značajna labela.

Za razliku od MPLS-TE-a koji koristi zasebne signalne protokole kako bi LSR-ovi međusobno izmijenili labele, unutar SR-a segmenti se oglašavaju putem *Intermediate System to Intermediate System* (IS-IS), *Open Shortest Path First* (OSPF) ili *Border Gateway Protocola* (BGP). Svaki od tih protokola ima definirane ekstenzije za distribuciju SID vrijednosti te se mogu pojedinačno aktivirati na usmjeritelju.

U kontekstu IP mreža, najčešće vrste segmenata su IGP segmenti [7], te postoje dvije vrste:

- Prefiks segmenti: Segment koji predstavlja neki IP prefiks unutar IP mreže. Ova vrsta segmenata se obično još dijeli na Segmente čvora (engl. *Node*) i *anycast* segmente. U ovom radu će se spominjati jedino segmenti čvora koji se najčešće odnose na Loopback (/32) IP adresu nekog usmjeritelja, što znači da je jednoznačan unutar SR domene, te je najčešće globalno poznat.
- Segmenti susjedstva: Segmenti koji predstavlja neko susjedstvo (tj.vezu) između 2 susjedna usmjeritelja. To su lokalni segmenti koji su poznati samo usmjeriteljima koji su povezani tom vezom. Usmjeritelji alociraju neku proizvoljnu dinamičnu vrijednost van SRGB-a koja ne mora biti jednoznačna unutar SR domene. Svaki usmjeritelj na toj vezi može imati vlastitu (drugačiju) oznaku te veze.

Primjer neke jednostavne SR domene sa pripadajućim SID vrijednostima za prefiks segmente i segmente susjedstva (gledano s strane usmjeritelja s manjim indeksom) se može vidjeti na Slika 2.1 gdje je svakom usmjeritelju dodijeljen globalni SID prema indeksu usmjeritelja, a veze su označene SID vrijednostima koje u realnom okruženju usmjeritelji dinamično dodjeljuju.



Slika 2.1 Skica SR domene sa SID vrijednostima Prefiks segmenata i segmenata susjedstva

2.2. SR politike i identifikacija SR politika

Kako bi se željeni promet usmjerio kroz SR domenu prema željenoj putanji i odredištu, potrebno je implementirati SR politike (engl. *policy*). To su skupovi pravila koji definiraju specifične rute i načine na koje će se promet usmjeravati kroz mrežu do svojeg konačnog odredišta kako bi se zadovoljili određeni zahtjevi.

Globalno unutar SR domene, svaka SR politika je jedinstveno označena izrazom (1).

$$(head_end, color, end_point) \quad (1)$$

Head-end označava usmjeritelj na kojem je kreirana SR politika, *color* je proizvoljna numerička vrijednost koja služi za razlikovanje više SR politika između *head-enda* i *end-pointa*, a *end-point* je odredište SR politike [8]. Bitno je napomenuti da jedna SR politika konfigurirana na *head-endu* se odnosi na mrežni promet od *head-enda* do *end-pointa*, ali ne i obrnuto. Drugim riječima, za dvosmjerni TE je potrebno konfigurirati SR politiku i na *end-pointu* prema *head-endu*, koja može ali i ne mora biti iste specifikacije. Time se omogućuje da promet u jednom smjeru ide jednom putanjom, a u drugom smjeru drugom putanjom, što se još zove i asimetrično usmjeravanje. Iako se u današnjim mrežama asimetrično usmjeravanje ne preferira ponajviše zbog *stateful* vatrozida i *Network Address Translation* (NAT) uređaja [10], u kontekstu interno konfiguriranih usluga u mreži nekog ISP-a može značajno pomoći u zaobilazanju jednostrano zagušenih ili (iako danas već rijetkih) asimetričnih veza.

2.3. SR putanje

Svaka SR politika se sastoji od jednog ili više kandidirajućih putanja (engl. *Candidate paths*) koja reprezentira neki specifičan način prosljeđivanja prometa od *head-enda* do *end-pointa* za tu SR politiku.

Svaka definirana kandidirajuća putanja unutar SR politike ima prioritet koji je izražen kao numerička vrijednost. Što je vrijednost viša, to je putanja većeg prioriteta.

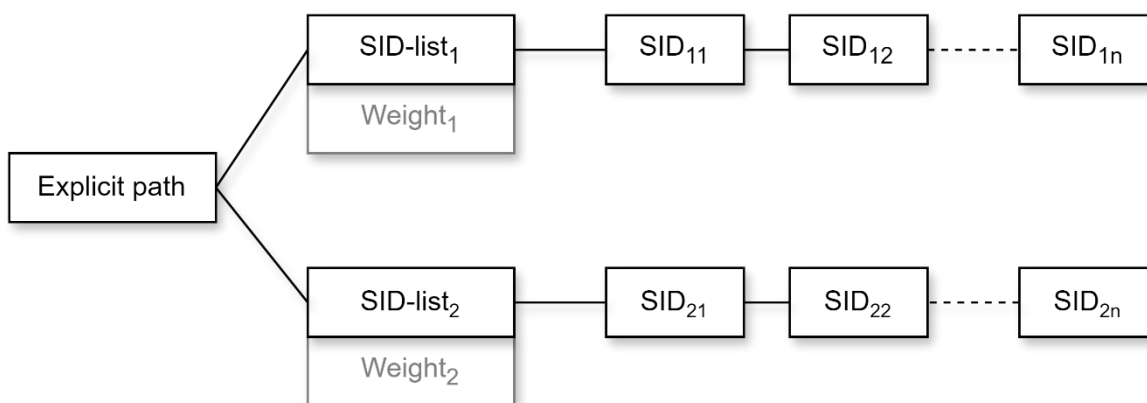
Aktivna putanja je ona kandidirajuća putanja koja je ispravna (engl. *valid*) i ona koja ima najviši prioritet. Ukoliko se na *head-end* usmjeritelju izbriše ili modificira aktivna putanja, dodaje nova ili modificira postojeća kandidirajuća putanja, proces selekcije aktivne putanje se ponovno pokreće [9]. SR politika se smatra ispravnom i aktivnom ako je barem jedan

kandidirajući put ispravan. Ako je SR politika neispravna (engl. *invalid*) onda ju usmjerenik neće koristiti, već će po zadanoj konfiguraciji usmjeravati pakete (koji bi inače išli nekom SR putanjom) koristeći npr. IGP najkraći put.

2.3.1. Eksplicitne putanje

Eksplicitna (kandidirajuća) putanja unutar SR politike je statički definirana jedna ili više SID lista gdje svaka lista ima pridruženu težinsku vrijednost (engl. *weight*).

Svaki pojedinačni SID unutar SID liste se može izraziti pomoću same SID vrijednosti, tj. vrijednosti labele ili pomoću nekog SID deskriptora koji se može razlučiti na SID, npr. *Loopback* IP adresa usmjerenika. Skica strukture može se vidjeti na Slika 2.2 ispod.



Slika 2.2 Skica strukture eksplicitne putanje

SID lista može biti u dva stanja:

- Ispravnom (engl. *valid*)
- Neispravnom (engl. *invalid*)

SID lista se nalazi u neispravnom stanju ako je prazna, ako *head-end* ne može razlučiti prvi SID unutar SID liste na jedan ili više sučelja i sljedećih skokova (engl. *hopova*) ili ako *head-end* ne može razlučiti bilo koji SID deskriptor na sam SID (tj. labelu). Ako ni jedan od ovih uvjeta nije zadovoljen, SID lista se smatra ispravnom i usmjerenik će ju koristiti ako je kandidirajuća putanja unutar koje je ta SID lista definirana ujedno i aktivna putanja.

Poput SID liste, eksplicitna putanja može isto biti u ispravnom ili neispravnom stanju. Kako bi eksplicitna putanja bila u ispravnom stanju, potrebno je da je barem jedna pripadajuća

SID lista bude u ispravnom stanju. U slučaju da unutar aktivne eksplicitne putanje postoje ispravne i neispravne SID liste, usmjerenje će za prosljeđivanje paketa koristiti samo ispravne SID liste. *Head-end* će prilikom svake promjene unutar mrežne topologije ponovno provjeravati ispravnost svake SID liste unutar svih konfiguriranih SR politika kojih se te promjene tiču [8].

Kao što se može vidjeti na Slika 2.2, svaka SID lista ima pripadajuću težinsku vrijednost koja je, ako se eksplicitno ne navede unutar konfiguracije, tvornički zadana na vrijednost 1. Kad unutar aktivne eksplicitne putanje postoji više SID lista koje su ispravne, prosljeđivanje paketa po pojedinačnim SID listama radi se proporcionalno njihovim težinskim vrijednostima, tj. prema sljedećoj formuli (2) [8]:

$$\frac{Težina_x}{\sum_{i=1...n} Težina_i} \quad (2)$$

Takav način prosljeđivanja paketa se zove *Weighted Equal-Cost Multi-Path routing* (WECMP). Primjer SR politike koja ima (aktivnu) eksplicitnu putanju i dvije SID liste s usmjerenjem „R1“ na usmjerenje „R4“ iz prethodnog primjera sa Slika 2.1 gdje se može vidjeti kako se paketi kreću dvjema putanjama različitih težina vidi se na Slika 2.3, a primjer konfiguracije SR politike na Cisco IOS XRv9000 usmjerenju koji predstavlja „R1“ usmjerenje sa slike, vidi se u Kod 2.1

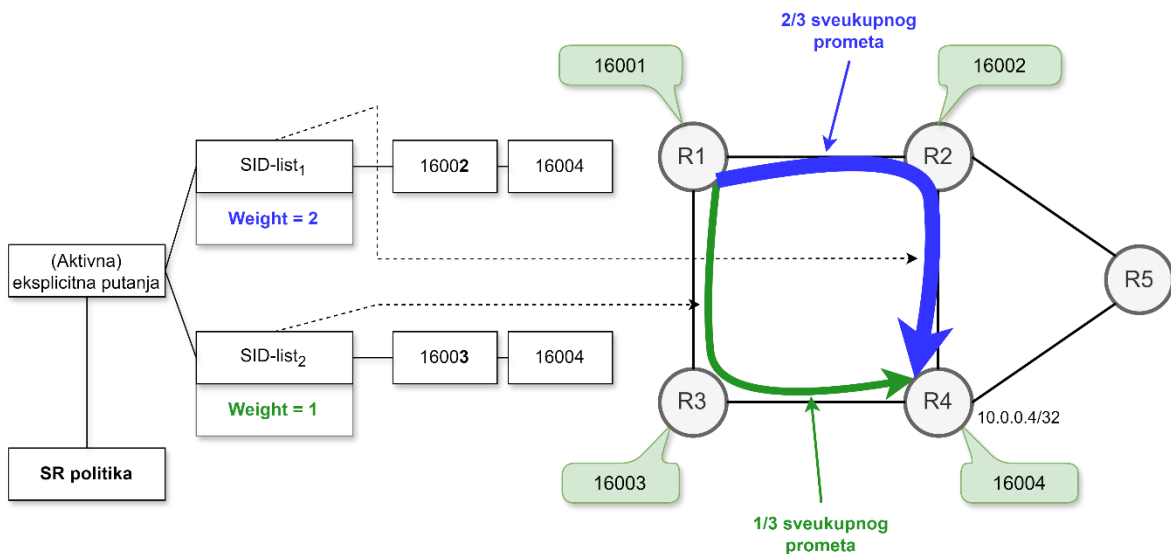
```
segment-routing
traffic-eng
policy TestPolicy
  color 10 end-point ipv4 10.0.0.4
  candidate-paths
  preference 100
  explicit segment-list SIDlist1
  weight 2
  !
  explicit segment-list SIDlist2
  weight 1
  !
```

```

segment-list name SIDlist1
  index 10 mpls label 16002
  index 20 mpls label 16004
!
segment-list name SIDlist2
  index 10 mpls label 16003
  index 20 mpls label 16004

```

Kod 2.1 Primjer konfiguracije dvije eksplicitne putanje sa različitim težinskim vrijednostima



Slika 2.3 Primjer WECMP-a na jednoj SR politici

U konfiguracijskom kodu usmjeritelja može se jasno uočiti kreirana politika s imenom `TestPolicy` gdje je definiran *color* s vrijednošću 10 i *end-point* „10.0.0.4“ što je *loopback* IP adresa usmjeritelja „R4“. Unutar politike definirana je jedna kandidirajuća putanja sa prioritetom 100, unutar koje se specificira korištenje dvije SID liste s pripadajućim težinskim vrijednostima 1 i 2. Ispod konfiguracije politike može se uočiti i sama konfiguracija SID lista, koje su u ovom slučaju indeksirane liste SID vrijednosti (tj. pripadajućih MPLS labela). Indeksni broj unutar definicije SID liste je proizvoljan, ali bitno je da inkrementalno prikazuje redoslijed SID-ova

2.3.2. Dinamične putanje

Za razliku od eksplicitnih putanja koje se statički definiraju prilikom konfiguracije SR politika, dinamičke putanje unutar SR politike je dinamično izračunata SID lista ili skup SID lista na temelju konfiguriranog cilja optimizacije i skupova ograničenja koje omogućuju automatsko prilagođavanje i optimizaciju usmjeravanja prometa na temelju stvarnih mrežnih uvjeta. Izračunavanje optimalnog puta obično radi *head-end Constrained Shortest Path First* (CSPF) algoritmom, no ukoliko *head-end* nema dovoljno informacija o topologiji, npr. u slučajevima poput više domena IGP-a unutar mreže ISP-a, izračun se delegira *Path Computation Elementu* (PCE). Svaki put kada *head-end* ili PCE prime dojavu o promjeni unutar mreže, radi se rekalkulacija optimalne dinamične putanje kako bi se putanja prilagodila novom stanju u mreži [9].

Usporedba MPLS-TE i SR-TE

Iako je klasični MPLS-TE omogućio funkcionalnost spojnih usluga unutar klasičnih IP mreža, a u isto vrijeme može rješavati optimizacijske probleme s ograničenjima koji kao rješenje daju putanju kojom će teći promet, *SR-native* algoritmi omogućuju ECMP-a i time bolje iskorištavaju mrežne resurse i smanjuju veličinu SID liste.

Razlike između MPLS-TE i SR-TE tehnologija su u tome što MPLS-TE zahtijeva posebne signalne protokole, dok SR-TE iskorištava IGP protokole koji su već implementirani u mrežu ISP-a. Obje tehnologije podržavaju FRR ali na svoj način. Kod MPLS-TE-a uvijek se izračunava sekundarna putanja kako bi se osigurala sigurnost od ispada, no zbog toga je u nekim slučajevima potrebna dvostruka rezervacija resursa RSVP-TE protokola što je neefikasno [11]. Za razliku od toga SR-TE, čim korišteni IGP protokol konvergira nakon ispada veze ili čvora, automatski će koristiti najbolju putanju koja je trenutno dostupna. Isto tako, ne postoji dodatni *overhead* kao kod MPLS-TE-a gdje svaki čvor mora u tablici prosljeđivanja spremati sekundarnu putanju do sljedećeg čvora. Kao što je i prije bilo spomenuto, SR-TE je napravljen kako bi iskoristio ECMP unutar mreže gdje god je to moguće, dok MPLS-TE ne podržava tu mogućnost, već jedini način kako bi se ECMP mogao realizirati jest da se konfiguriraju 2 tunela koji će ići određenom putanjom kako bi iskoristio više veza. Naravno, SR-TE značajno pojednostavljuje kompletnu konfiguraciju i implementaciju, skalabilniji je jer ne zahtijeva posebne signalne protokole i potrebna je puno manja tablica prosljeđivanja za njegovu implementaciju, pogotovo u većim mrežama. Skraćeni prikaz osnovnih razlika može se vidjeti u Tablica 2.1

Tablica 2.1 Usporedba MPLS-TE i SR-TE tehnologija, Izvor: preuzeto iz [12]

	SR-TE	MPLS-TE
Razmjena i signalizacija labela	IGP	LDP + RSVP-TE
FRR	IGP	IGP + RSVP-TE
Dodatni overhead za FRR funkcionalnost	Ne	Da
ECMP	Da	Ne
Skalabilnost	Dobra	Loša
Kompliciranost održavanja i otklanjanja kvarova	Jednostavno	Kompliciranije

Metrike

SR-TE podržava tri vrste metrika po kojoj se može raditi optimizacija:

1. IGP metrika
2. Metrika kašnjenja
3. TE metrika

Prva na listi, IGP metrika, je korištenje metrike samog IGP protokola. Ukoliko ne postoji konfiguriranih ograničenja unutar dinamične putanje koja koristi IGP metriku, paketi koji će prolaziti kroz tu SR politiku (od *head-enda* do *end-pointa*) ići će istom putanjom kojom bi išli da se koristilo klasično usmjeravanje paketa na IP sloju.

Sljedeća metrika koju SR-TE može koristiti kao optimizacijsku varijablu je kašnjenje, no jedini parametar koji može koristiti je minimalno kašnjenje. To znači da varijable poput prosječnog kašnjenja, maksimalnog kašnjenja i *jittera* ne igraju ulogu u izračunavanju najoptimalnijeg puta koristeći ovu vrstu metrike. Te ostale varijable, a pogotovo *jitter*, su vrlo ovisne o trenutnim uvjetima na linku i mogu se mijenjati u kratkim vremenskim periodima te se zato ne uzimaju u obzir u kontekstu optimizacije usmjeravanja, već se uzima isključivo minimalno kašnjenje. Razlog tome je što se varijabla minimalnog kašnjenja može korelirati s propagacijskim kašnjenjem veze tj. zapravo služi kao mjera propagacijskog kašnjenja. Ova metrika može opisivati kašnjenje u jednom ili oba smjera, predstavlja svojstvo same topologije i može detektirati promjene poput npr. prebacivanja DWDM linkova na drugu nit zbog fizičkog oštećenja. Fizički Cisco IOS XRv9000 usmjeritelji sve prije spomenute varijable mjere i računaju među sobom na sučeljima koristeći RFC 6374

pakete [13], a vrijednost minimalnog kašnjenja uključuju kao parametar u IGP poruke. Po zadanim postavkama, Cisco usmjeritelj šalje svakih 3 sekunde upit prema drugom usmjeritelju, nakon svakih 30 sekundi izračunava sve varijable, a nakon svakih 120 sekundi, ukoliko se minimalno kašnjenje promijenilo za više od 10% prethodne vrijednosti, šalje se IGP poruka sa novom vrijednošću na sve susjede. Cisco IOS XRv9000 usmjeritelji mogu mjeriti kašnjenje u jednom ili oba smjera, ali za mjeru kašnjenja u jednom smjeru potrebna je sinkronizacija satova između svih usmjeritelja [14]. Primjer konfiguracije mjerenja kašnjenja u oba smjera na jednom sučelju može se vidjeti na konfiguracijskom Kod 2.2

```
performance-measurement
interface GigabitEthernet0/0/0/0
  delay-measurement
  !
  !
  delay-profile interfaces default
  probe
  measurement-mode two-way
  !
  !
  !
```

Kod 2.2 Primjer konfiguracije mjerenja kašnjenja

Zadnja podržana metrika je TE metrika. Ta vrsta metrike potječe iz MPLS-TE tehnologije gdje je osnovna zamisao bila da se omogući ISP-u da konstruira LSP (za neki promet) baziran na drugoj vrsti metrike koja može predstavljati neke druge mjere osim one koje je ISP postavio kao IGP metriku. Npr. TE metrika mogla je predstavljati statički definirane preferencije ISP-a za određene veze. Time se omogućuje da ISP u kontekstu CSPF računanja LSP-a ima dvije TE topologije, gdje je svaka bazirana na jednoj metrici [15]. Na Cisco IOS XRv9000 uređajima, unutar SR-TE-a moguće je definirati TE metriku na svakom od sučelja kao što se može vidjeti na primjeru Kod 2.3, a ako sučelje nema izričito definiranu TE metriku, prilikom izračuna SID liste za neku SR politiku, vrijednost TE metrike za to sučelje uzimati će se kao IGP metrika tog sučelja.

```
segment-routing
traffic-eng
interface GigabitEthernet0/0/0/0
metric 25
!
!
!
```

Kod 2.3 Primjer dodjeljivanja TE metrike na sučelje

Margine

Prilikom izračunavanja dinamične SID liste, CSPF proces će uvijek vratiti listu koja je maksimalno optimizirana po metrici koja je odabrana u SR politici. No ponekad u mreži postoje putanje koje su veoma sličnih karakteristika te se u realnom svijetu mogu neznajčajno razlikovati (npr. 400 μ s razlike minimalnog kašnjenja), a ipak su definirane različite vrijednosti metrika na usmjeriteljima za te putanje. U tom slučaju bi promet koji ide po toj definiranoj SR politici išao isključivo po najboljoj putanji.

Specificiranje margina unutar SR politike omogućuje da se relaksira striktnost optimizacije te se preferira više ECMP i kraća SID lista. U slučaju kada se vrijednosti metrika mijenjaju na vezama, ako postojeća putanja s manjom SID listom neke SR politike ima metriku koja je manja ili jednaka trenutno najboljoj mogućoj metrici do odredišta zbrojenoj s vrijednosti margine, putanja SR politike se neće mijenjati.

Margine se mogu specificirati kao apsolutna vrijednost ili kao relativna vrijednost. Apsolutna vrijednost je prirodan broj koji numerički specificira kolika je margina za neku odabranu vrstu metrike. Relativna vrijednost specificira marginu u postocima od trenutno najbolje vrijednosti metrike do odredišta. Npr. postojeća putanja s manjom SID listom ima kašnjenje od 110ms, te se pojavi nova putanja koja omogućuje kašnjenje od 100ms. Ako je specificirana margina 10% na toj SR politici, putanja se neće optimizirati jer trenutna putanja je još uvijek unutar margine [16].

Ograničenja

Unutar SR-TE-a postoje razne mogućnosti ograničenja kod izračuna putanja koja omogućuju ISP-u bolje upravljanje prometom i putanjama kojima će određeni promet ići. Iz razloga što se mogućnosti ograničenja razlikuju između verzija *firmwarea* Cisco IOS XRv9000 usmjeritelja, u ovom radu neće se spominjati sve mogućnosti ograničenja već će

se fokusirati na ona osnovna koja su podržana u svakoj verziji. Ta osnovna ograničenja se svode na afinitet (engl. *affinity*) i disjunktne puteve.

Afinitet je vrsta oznake koja se može dodijeliti nekoj vezi. Svaka veza može imati više ili niti jednu oznaku. Unutar SR politike može se specificirati 3 tipa logičkih mogućnosti za afinitete:

- `Include-all`
- `Include-any`
- `Exclude-any`

`Include-all` specificira da sve veze kojom će prolaziti promet u SR politici moraju imati navedene afinitete koji se napišu uz `include-all` opciju. `Include-any` opcija zahtijeva da sve veze imaju barem jedan afinitet od liste definiranih afiniteta (lista može biti samo jedan afinitet), dok `exclude-any` opcijom se izbjegavaju sve veze koje imaju barem jedan afinitet od definirane liste. Ova vrsta ograničenja otvara vrlo velike mogućnosti unutar mreže ISP-a, jer ISP, ovisno o tome koliko detaljno želi definirati afinitete, može vezama pridruživati oznake koje ih opisuju prema geografskim svojstvima (npr. transkontinentalne i međudržavne oznake), svojstvima medija (npr. parice, SM i MM optičke niti itd.), oznake uređaja kojima prolaze (npr. fizički DWDM ili komutator) i mnoge druge. Time se SR politika može veoma točno definirati, bilo prema zahtjevima ili potrebama korisnika.

Sve vrste afiniteta koje ISP planira koristiti unutar svoje mreže na Cisco IOS XRv9000 usmjeriteljima se moraju globalno i jednoznačno unutar SR domene definirati na usmjeriteljima koji ih koriste unutar SR politika i koji su povezani vezama označenim nekim afinitetima, jer afiniteti su zapravo bitovi unutar mape bitova (engl. *bitmap*) te svaki bit predstavlja jedan afinitet. Tu mapu bitova IGP protokol oglašava zajedno s ostalim informacijama kako bi *head-end* ili PCE mogli znati i ispravno izračunati putanju za neku SR politiku [18]. Primjer mape afiniteta (engl. *affinity map*) koja specificira oznake za vrstu medija se može vidjeti u Kod 2.4 ispod.

```
segment-routing
traffic-eng
  affinity-map
    name Optics bit-position 0
    name Coax bit-position 1
```

```
name TwistedPair bit-position 2
```

Kod 2.4 Primjer konfiguracije mape afiniteta na Cisco uređajima

Drugo bitno ograničenje kod računanja putanje za SR politiku su disjunktni putovi. Kod definiranja SR politike na Cisco IOS XRv9000 uređajima moguće je specificirati `group-id` unutar `disjoint-path` ograničenja. Time se unutar mreže svi čvorovi i veze kojom prolazi ta SR politika označavaju s konfiguriranim `group-id` brojem. Prilikom definiranja neke druge SR politike (između bilo koja dva usmjeritelja) konfiguracijom istog `group-id` broja unutar `disjoint-path` ograničenja mogu se zaobići čvorovi (engl. *node*) ili veze (engl. *link*) kojom prolazi prva SR politika. Time se omogućava automatska fizička odvojenost dvaju SR politika unutar mreže ISP-a [17].

Kompletan primjer jednostavne konfiguracije Cisco IOS XRv9000 usmjeritelja koja sadrži dvije SR politike prema istom *end-pointu* ali različitih *color* vrijednosti, konfigurirane s različitim metrikama (IGP i TE), definiranim apsolutnim marginama s vrijednosti 10, definiranom grupom 1 disjunktnih veza može se vidjeti u konfiguracijskom Kod 2.5. Skica mreže slična onoj iz Slika 2.1 vidi se na Slika 2.4 gdje se mogu uočiti putanje dvaju politika koje su međusobno disjunktna te kako jedna putanja prve SR politike definirane u konfiguracijskom Kod 2.5 zaobilazi vezu označenom `Coax` afinitetom iako je kumulativna metrika putanje bolja ako ide tom vezom. Bitno je napomenuti da usmjeritelji R1, R2 i R6 moraju imati definirani afinitet `Coax`, ali je preporučljivo da svi afiniteti uvijek budu definirani na svakom usmjeritelju unutar iste SR domene.

```
segment-routing
traffic-eng
affinity-map
  name Optics bit-position 0
  name Coax bit-position 1
  name TwistedPair bit-position 2
!
policy customerPathBlue
  color 1 end-point ipv4 10.0.0.6
  candidate-paths
    preference 100
    dynamic
```

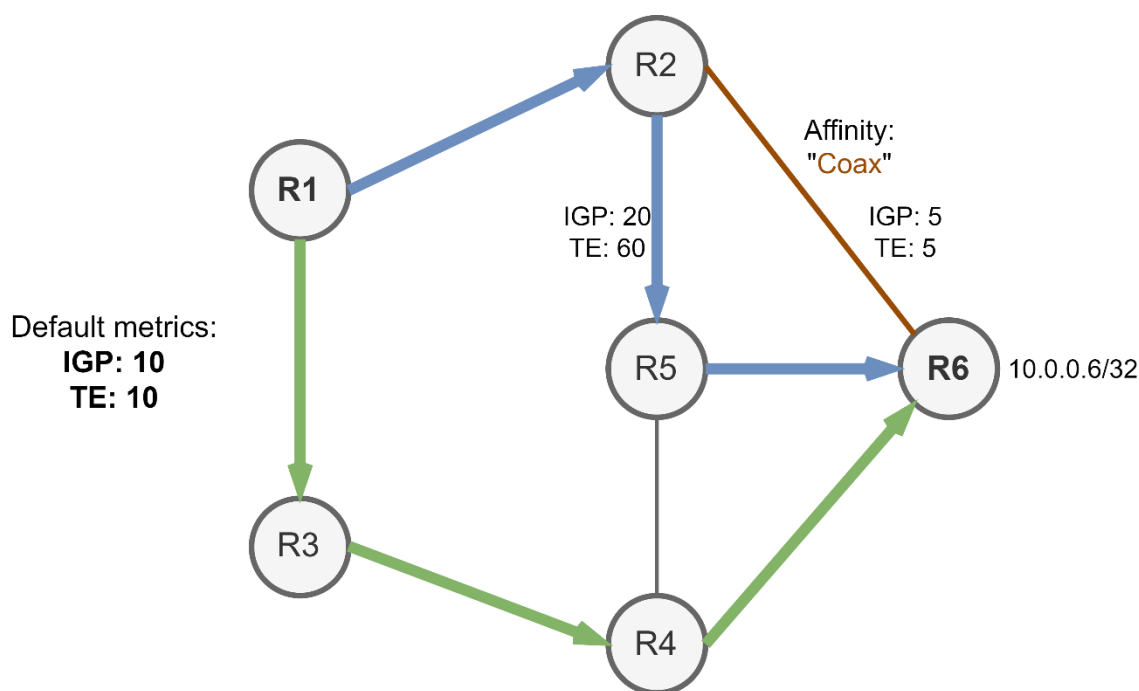


```

pcep
!
metric
  type igp
  margin absolute 10
!
!
constraints
  affinity
    exclude-any
      name Coax
    !
  !
  disjoint-path group-id 1 type link
!
!
!
!
policy customerPathGreen
  color 2 end-point ipv4 10.0.0.6
  candidate-paths
    preference 100
    dynamic
      pcep
      !
      metric
        type te
        margin absolute 10
      !
      !
      constraints
        disjoint-path group-id 1 type link

```

Kod 2.5 Primjer kompletne konfiguracije dvaju SR politika s marginama, ograničenjima i disjunktним putevima



Slika 2.4 Skica putanja dvaju SR politika definiranim u konfiguracijskom kodu 2.5

2.4. PCE

PCE je entitet (komponenta, aplikacija ili mrežni čvor) koji je sposoban izračunati putanju za neki mrežni promet uzimajući u obzir dana ograničenja, a komunicira s *Path Computation Clientima* (PCC) preko PCE Protocola (PCEP) [19][20]. Iako je unutar Cisco IOS XRv9000 *firmwarea* PCE funkcija integrirana, može biti realiziran i na dedicanom serveru. U kontekstu SR-a, služi za izračun putanja za SR politike kada *head-end* nema sve dostupne informacije o topologiji do *end-pointa* (npr. multi domenske mreže) ili kada je potrebno izračunati putanje uz *disjoint-path* ograničenja [21]. Konfiguracija SR PCE-a na Cisco IOS XRv9000 uređaju kao i definiranje jednog *peera* s IP adresom 10.0.0.1 vidi se u Kod 2.6, a primjer dijela konfiguracije jednog usmjeritelja koji služi kao PCC se vidi u Kod 2.7. Ovdje je bitno spomenuti kako se unutar konfiguracije PCC-a može dodati više PCE-a sa različitim prioritetima, jer unutar mreže ne mora biti prisutan samo jedan PCE. U slučaju da PCE sa najvećim prioritetom postane nedostupan, PCC automatski delegira daljnje računanje putanja sljedeće definiranom PCE-u po prioritetu. U konfiguracijskom Kod 2.5 opcijom *PCEP* se izračun dinamične putanje delegirao PCE-u što je u primjeru na Slika 2.4 omogućilo izračun disjunktnih putanja.

```
pce
address ipv4 10.0.0.12
api
!
peer ipv4 10.0.0.1
```

Kod 2.6 Konfiguracija PCE-a

```
segment-routing
traffic-eng
pcc
source-address ipv4 10.0.0.1
pce address ipv4 10.0.0.12
precedence 100
```

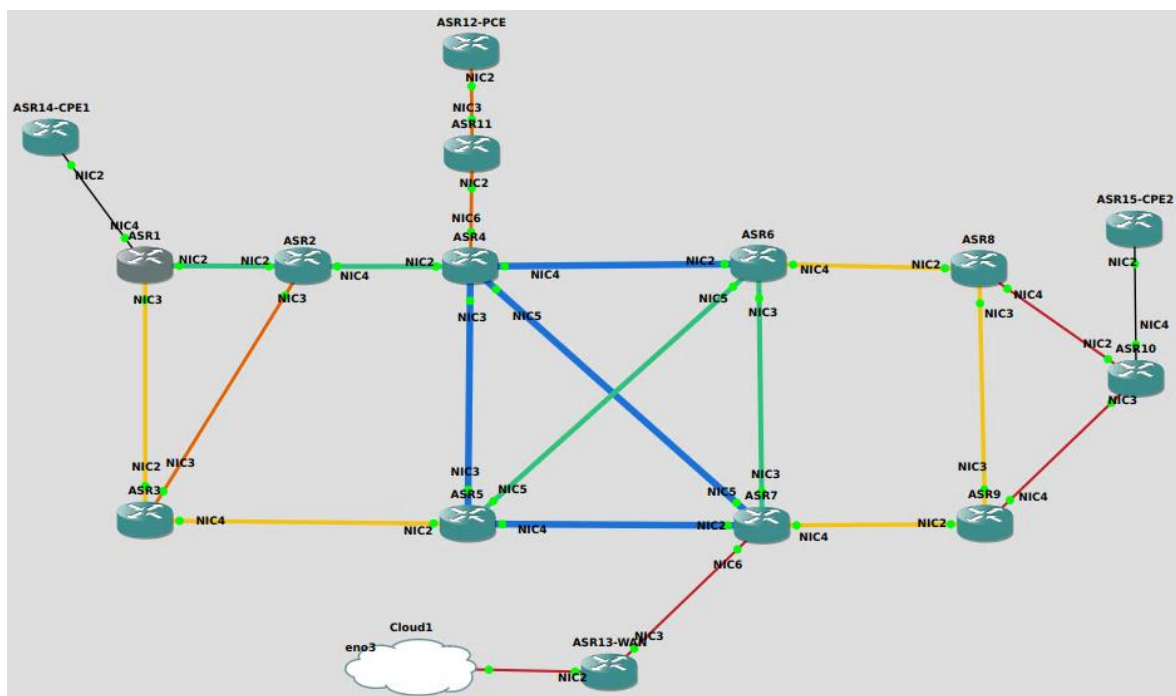
Kod 2.7 Konfiguracija PCC-a

3. Simulacija i konfiguracija

Za potrebe ovog rada izrađena je kompletna simulacija mreže ISP-a, tj. usmjeritelja i njihovih međusobnih veza, čime se u fokus stavlja upravljanje mrežom na trećem sloju OSI modela uključujući MPLS i SR za ostvarenje željenih zahtjeva i usluga.

3.1. Simulator i virtualizacija







Za potrebe rada korišten je *open source* program *Graphical Network Simulator-3* (GNS3) koji omogućuje emulaciju fizičkih uređaja i njihovih *firmwarea*, ali i simulaciju raznih mrežnih komponenti poput preklopnika ili jednostavnih *hubova* koji ne pokreću neki operativni sustav fizičkog uređaja, već se njihova funkcionalnost implementira programskim kodom unutar GNS3-a [22]. Unutar GNS3-a emulirano je sveukupno 15 Cisco IOS XRv9000 usmjeritelja od kojih njih 12 služe kao usmjeritelji unutar mreže ISP-a, dva kao *Customer Premises Equipment* (CPE), te dodatno jedan kao PCE. Svaki od njih pokreće istu verziju (7.7.2) *firmwarea* [32][33].



Slika 3.1 Mrežna topologija simulirane mreže ISP-a unutar GNS3

Na Slika 3.1 može se vidjeti slika mrežne topologije korištene u ovom radu kao i više različito obojanih veza između usmjeritelja. Svaka boja predstavlja propusnost veze između dva usmjeritelja, čija su svojstva detaljno opisana u Tablica 3.1. Bitno je napomenuti kako unutar GNS3 koristeći IOS XRv9000 virtualne usmjeritelje, fizička sučelja su uvijek gigabitna, tj. među njima nema fizičkih razlika, već se radi isključivo o gigabitnim sučeljima na kojima su konfigurirane različite metrike i oznake kako bi prikazivala različite kapacitete. Isto tako, zbog procesorske zahtjevnosti simulacije Cisco IOS XRv9000 usmjeritelja, ne postoji mogućnost saturiranja sučelja i izrada simulacije u kojem bi prolazio mrežni promet kako bi se prikazalo stvarno stanje mreže ISP-a.

Tablica 3.1 Virtualni kapaciteti veza u mrežnoj simulaciji

Kapacitet veze	Boja	Boja (vizualno)
nedefinirano	crna	
1Gbit/s	crvena	
10Gbit/s	narančasta	
25Gbit/s	tamnožuta	
40Gbit/s	zelena	
100Gbit/s	plava	

3.2. Konfiguracija usmjeritelja

Svaki od usmjeritelja ima osnovnu konfiguraciju ISIS-a i SR-TE-a koja je spremna za kreiranje SR politika. Konfiguracija je rađena od samog početka te nema primijenjenih tehnologija koje bi se mogle naći unutar stvarne mreže ISP-a, ali isto tako sva konfiguracija napravljena unutar simulacije može koegzistirati s trenutno korištenim tehnologijama unutar stvarnih mreža. Sljedeći primjeri konfiguracije koji predstavljaju generalnu konfiguraciju svih usmjeritelja, uzeti su sa *ASR1* usmjeritelja koji je spojen prema *ASR2,ASR3* i *ASR14-CPE* usmjeriteljima kao što se vidi na Slika 3.1.

3.2.1. Konfiguracija sučelja i ISIS-a

U Kod 3.1 nalazi se konfiguracija ISIS-a i sučelja na *ASR1* usmjeritelju. U prva dva reda definirano je *loopback* sučelje i njegova IP adresa. Općenito na svim usmjeriteljima sintaksa

loopback IP adrese je takva da prvi usmjeritelj počinje s privatnom IP adresom 10.0.0.1/32, a svi sljedeći usmjeritelji imaju IP adrese inkrementalno uvećane za jedan. Na sučeljima koja služe za međusobno povezivanje usmjeritelja (u ovom slučaju GigabitEthernet0/0/0/0 i GigabitEthernet0/0/0/1) konfigurirane su IP adrese u privatnom rasponu 192.168.X.Y, gdje je svaka veza između dva usmjeritelja u vlastitoj /24 podmreži, a adresa *hosta* je 1 ili 2, ovisno koji usmjeritelj ima manji broj *loopback* IP adrese. Iako se u realnim mrežama koriste /31 podmreže za uspostavljanje susjedstva, u ovom radu koristile su se /24 podmreže radi lakše sintakse i snalaženja. Nadalje, na svakom usmjeritelju konfiguriran je ISIS proces i svi usmjeritelji su unutar iste *level 2* ISIS domene. Sljedeća linija konfiguracijskog koda, `distribute link-state`, omogućuje prenošenje topologije dobivene ISIS procesom u SR-TE topološku bazu podataka [23]. Ova komanda, ukoliko se za svaku SR politiku koristi PCE, nije potrebna na svakom PCC-u, ali ako PCC želi sam moći izračunati putanju tj. SID listu za lokalno definiranu SR politiku, onda ju treba omogućiti. Sljedećih par linija se odnosi na IPv4 *unicast* promet. Na prvoj liniji je definirano korištenje isključivo proširenih (engl. *wide*) metrika kako bi ISIS mogao procesuirati novije *Type Length Value* (TLV) poruke sa proširenim vrijednostima metrika koje su potrebne radi funkcioniranja nekoliko funkcionalnosti unutar TE-a [24]. Sljedeća linija konfigurira oglašavanje isključivo IP adresa koje pripadaju pasivnim sučeljima, a ona nakon definira sučelje na koje će se odnositi oglašavani *prefix* SID za taj usmjeritelj. Zadnja linija unutar `address-family ipv4 unicast` sekcije konfigurira usmjeritelj da taj promet usmjeri na MPLS prosljeđivanje paketa, sa preferiranim prosljeđivanjem koristeći SR (ako se npr. koristi i LDP) [25]. Ostatak linija odnosi se na konfiguraciju sučelja unutar ISIS-a. *Loopback0* sučelje je definirano kao pasivno, i definiran je `prefix-SID index 1`, što znači da će globalna oznaka SID-a za ovo sučelje i IP adresu biti SRGB baza zbrojena s brojem *indeks* vrijednosti (U ovom slučaju, radi se o SID-u 16001). Ostala sučelja konfigurirana su kao point-to-point jer svako sučelje je spojeno samo s jednim usmjeriteljem.

```
interface Loopback0
  ipv4 address 10.0.0.1 255.255.255.255
!
interface GigabitEthernet0/0/0/0
  description ASR2-link
  ipv4 address 192.168.1.1 255.255.255.0
```

```

!
interface GigabitEthernet0/0/0/1
  description ASR3-link
  ipv4 address 192.168.2.1 255.255.255.0
!

router isis 1
  is-type level-2-only
  net 49.0001.0100.0000.0100.00
  distribute link-state
  log adjacency changes
  address-family ipv4 unicast
    metric-style wide
  advertise passive-only
  router-id Loopback0
  segment-routing mpls sr-prefer
!
interface Loopback0
  passive
  address-family ipv4 unicast
    prefix-sid index 1
!
!
interface GigabitEthernet0/0/0/0
  point-to-point
  address-family ipv4 unicast
    metric 25
!
!
interface GigabitEthernet0/0/0/1
  point-to-point
  address-family ipv4 unicast
    metric 40

```

Kod 3.1 Konfiguracijski kod ASR1 usmjeritelja

Metrika na sučeljima koja je definirana u konfiguracijskom Kod 3.1 reprezentira njihovu propusnost u odnosu prema brzini od 1Tbit/s. Izračun metrike radi se po formuli (3) gdje IGP_i predstavlja IGP metriku za sučelje i , a BW_i propusnost tog sučelja u megabitima po sekundi.

$$IGP_i = \frac{1}{BW_i [Mbit/s]} * 1000000[Mbit/s] \quad (3)$$

Ponekad veza između dva usmjeritelja nije fizičko sučelje već neki VLAN unutar fizičkog sučelja, *bundle-ether* ili npr. iznajmljena *Virtual Private Lan Service* (VPLS) usluga s ograničenom brzinom kojom se realizira povezanost dva usmjeritelja. U tom slučaju se u formulu ne uvrštava propusnost fizičkog sučelja kojim je jedan usmjeritelj povezan s drugim, već maksimalna idealna brzina koja se može postići na najnižem sloju između ta 2 usmjeritelja. Za tu potrebu ponekad je dobro iskoristiti *bandwidth* vrijednost koja je unutar Cisco IOS XRv9000 zapisana u Kbit/s te se može pojedinačno konfigurirati za svako (virtualno ili fizičko) sučelje. U tom slučaju vrijednost *bandwidtha* se pomnoži s 1000 i uvrsti u formulu. Ova ideja računanja IGP metrike daje prioritet vezama koje imaju najviše kapaciteta.

3.2.2. Konfiguracija SR-TE-a

U konfiguracijskom Kod 3.2 prikazana je konfiguracija SR dijela konfiguracije na ASR1 usmjeritelju. Odmah u drugoj liniji može se vidjeti definicija SRGB-a. Nakon toga može se vidjeti definirana mapa afiniteta koji se koriste unutar ove simulacije. Ovo je jedan od važnijih dijelova konfiguracije jer definira osnovne afinitete koji omogućuju konkretnije upravljanje prometom. Osim afiniteta koji označuju međudržavne, međukontinentalne veze te veze koje su trenutno pod radovima, ovdje se definiraju i afiniteti propusnosti veza. Prva vrsta je jednoznačna oznaka propusnosti veze, dok druga vrsta je oznaka da je propusnost veze veća ili jednaka vrijednosti oznake. Nakon definicije mape afiniteta slijedi pridruživanje tih afiniteta na sučelja s kojima je usmjeritelj povezan s drugim usmjeriteljima. Npr. Na sučelje `GigabitEthernet0/0/0/0` koje predstavlja sučelje s propusnošću 40Gbit/s će prema ovoj mapi afiniteta imati dodijeljene afinitete 40G (jer je to 40Gbit/s sučelje) te 100M+, 1G+, 10G+, 25G+, 40G+ jer je propusnost sučelja veća ili jednaka tim oznakama.

Na kraju konfiguracije omogućena je opcija `te-latency` koja omogućava *ECMP-aware* izračun putanje kada se koristi TE metrika, a nakon toga, kao i u konfiguracijskom Kod 2.7, omogućen je PCC, definirana je izvorišna adresa koja će se koristiti pri komunikacijom s PCE-om, te IP adresa i preferencija PCE-a.


```

segment-routing
global-block 16000 17000
traffic-eng
affinity-map
name 100M bit-position 0
name 1G bit-position 1
name 10G bit-position 2
name 25G bit-position 3
name 40G bit-position 4
name 100G bit-position 5
name 100M+ bit-position 6
name 1G+ bit-position 7
name 10G+ bit-position 8
name 25G+ bit-position 9
name 40G+ bit-position 10
name 100G+ bit-position 11
name Maintenance bit-position 12
name Transnational bit-position 13
name Transcontinental bit-position 14
!
interface GigabitEthernet0/0/0/0
affinity
name 40G
name 1G+
name 10G+
name 25G+
name 40G+
name 100M+
!
!
interface GigabitEthernet0/0/0/1
affinity
name 25G
name 1G+
name 10G+
name 25G+
name 100M+
!
!
te-latency

```

```
pcc
  source-address ipv4 10.0.0.1
  pce address ipv4 10.0.0.12
  precedence 100
!
!
!
!
```

Kod 3.2 SR dio konfiguracijskog koda ASR1 usmjeritelja

4. Upravljačka skripta

U sljedećim potpoglavljima opisuje se struktura upravljačke skripte, preduvjeti konfiguracije mreže za ispravno funkcioniranje te komponente i mogućnosti upravljačke skripte koje omogućavaju inovativno upravljanje SR politikama i metrikama unutar mreže kako bi se omogućilo dinamično upravljanje putanjama.

4.1. Osnovna struktura i preduvjeti

Programski kod upravljačke skripte pisan je u pythonu i podijeljen je u nekoliko segmenata, tj. funkcionalnosti za upravljanje mrežom i optimizacijom radi ostvarenja zahtjeva usluga. Skripta koristi nekoliko javno dostupnih programskih biblioteka i alata koji se mogu besplatno dohvatiti i instalirati pomoću *pip package managera*, a puna lista se nalazi u privitku ovog rada. Skripta prije ponude opcija korisniku se spaja preko *Secure Shella* (SSH) protokola na PCE čija se IP adresa zajedno s pristupnim podacima (korisničko ime i lozinka) definira statički unutar skripte u *dictionary* varijabli `pce_config`. Nakon spajanja skripta dohvaća IP adrese svih konfiguriranih i trenutno spojenih PCCa na PCE te ih sprema lokalno u zasebne datoteke `pce_config.txt` i `pce_peers.txt`. Skripta pretpostavlja da su unutar konfiguracije specificirane iste IP adrese PCCa s kojih PCCi pristupaju PCEu. Sukladno tome, ako se te dvije liste razlikuju (što može biti zbog npr. kvara usmjeritelja) skripta upozorava korisnika te ispisuje razlike. Isto tako, skripta pretpostavlja da se preko tih IP adresa može pristupiti usmjeriteljima kojima pripadaju, tj. da su to ujedno i *management* adrese

Glavna (engl. *main*) funkcija korisniku omogućava preko *Command Line Interfacea* (CLI) odabir nekoliko funkcionalnosti skripte, tj. mogućnosti:

- Kreiranje *Xconnect* P2P servisa
- Brisanje *Xconnect* P2P servisa
- Dohvaćanje topologije i izračunavanje TE metrika
- Izračunavanje margina za SR politike s aktivnom TE dinamičnom putanjom
- Pokretanje automatskog pozadinskog dohvaćanja topologije, izračunavanje TE metrika i izračunavanje margina za SR politike s aktivnom dinamičnom putanjom

Za svaku od ponuđenih mogućnosti napisana je zasebna funkcija unutar datoteke kako bi se osigurala modularnost programskog koda. Sve SSH konekcije koje se otvore prema usmjeriteljima se spremaju u globalnu varijablu kako bi se izbjegao nepotrebn *overhead* za stalno uspostavljanje novih konekcija prema usmjeriteljima kojima se često pristupa.

4.2. Kreiranje i brisanje Xconnect P2P servisa

Jedan od glavnih dijelova skripte odnosi se na potpomognuto interaktivno kreiranje *Layer 2 Virtual Private Network* (L2VPN) usluge. Kada korisnik odabere tu opciju (1) poziva se funkcija `xconnect_setup`. Odmah na početku funkcije korisnika se traži da upiše (*loopback*) IP adrese usmjeritelja između kojih želi ostvariti L2VPN uslugu. Skripta provjerava nalaze li se upisane IP adrese unutar datoteke `pce_peers.txt` te ako se ne nalaze, korisniku se ispisuje upozorenje (jer ovisno o implementaciji i konfiguraciji PCC usmjeritelja može biti da PCE nije potreban za izračun putanje). Skripta se spaja na upisane IP adrese i dohvaća sva trenutno postojeća sučelja na usmjeriteljima koja ispisuje korisniku poredana rednim brojevima i traži upis rednog broja sučelja kojeg korisnik želi iskoristiti za ostvarenje usluge.

Nakon dobivenih sučelja, poziva se funkcija `create_sr_policy` koja služi za kreiranje SR politike u smjeru prvog usmjeritelja čija je IP adresa upisana prema drugom. Unutar ove funkcije korisnika se traži da unese ime SR politike, nakon čega upisuje sve vrste kandidirajućih putanja i njihove preference. Korisnik može kreirati eksplicitne putanje zajedno s njihovom *weight* vrijednosti ili dinamične putanje s definiranim marginama i ograničenjima. Odabir *color* oznake radi se automatski unutar skripte na način da dohvaćaju sve *color* vrijednosti prema istom *end-pointu* (drugom usmjeritelju), te se uzima sljedeća slobodna vrijednost. Primjer ispisa skripte i unosa podataka za kreiranje SR politike s dvije eksplicitno definirane SID liste unutar jedne kandidirajuće putanje i jedne dinamične putanje s IGP metrikom i relativnom marginom 2, koja izbjegava linkove koji su označeni s oznakama 100M, 1G ili Maintenance te je dodjeljena u disjunktnu grupu 2 vidi se u ispisnom Kod 4.1

```
Choose an action:  
[1] Create Xconnect P2P service  
[2] Delete Xconnect P2P service
```

```
[3] Get current topology and TE data and apply calculated TE
metrics
[4] Calculate and update margins for SR policies with active
TE dynamic path
[5] Run background TE metric and margin calculation
[0] Exit
Enter your choice: 2
Enter IP address of the first router: >? 10.0.0.1
Enter IP address of the second router: >? 10.0.0.10
[1] Fri
[2] Intf
[3] Name
[4] Lo0
[5] Nu0
[6] Mg0/RP0/CPU0/0
[7] Gi0/0/0/0
[8] Gi0/0/0/1
[9] Gi0/0/0/2
[10] Gi0/0/0/3
[11] Gi0/0/0/4
[12] Gi0/0/0/5
Choose interface for Xconnect on 10.0.0.1 (1-12): >? 9
[1] Fri
[2] Intf
[3] Name
[4] Lo0
[5] Nu0
[6] Mg0/RP0/CPU0/0
[7] Gi0/0/0/0
[8] Gi0/0/0/1
[9] Gi0/0/0/2
[10] Gi0/0/0/3
[11] Gi0/0/0/4
[12] Gi0/0/0/5
Choose interface for Xconnect on 10.0.0.10 (1-12): >? 9
Xconnect P2P setup information:
Router 1 (10.0.0.1): Interface Gi0/0/0/2
Router 2 (10.0.0.10): Interface Gi0/0/0/2
Enter SR policy name: >? XconnTest
Enter preference for the path (or 'done' to finish): >? 100
```

```
Choose path type for preference 100 (explicit/dynamic): >?
explicit
Enter weight for the sidlist: >? 1
Enter segment type (mpls label, mpls adjacency, or 'done' to
finish sidlist): >? mpls adjacency
Enter value: >? 10.0.0.3
Enter segment type (mpls label, mpls adjacency, or 'done' to
finish sidlist): >? mpls label
Enter value: >? 16010
Enter segment type (mpls label, mpls adjacency, or 'done' to
finish sidlist): >? done
Create another sidlist for this preference? (y/n): >? y
Enter weight for the sidlist: >? 5
Enter segment type (mpls label, mpls adjacency, or 'done' to
finish sidlist): >? mpls label
Enter value: >? 16010
Enter segment type (mpls label, mpls adjacency, or 'done' to
finish sidlist): >? done
Create another sidlist for this preference? (y/n): >? n
Enter preference for the path (or 'done' to finish): >? 200
Choose path type for preference 200 (explicit/dynamic): >?
dynamic
Choose metric type (IGP, latency, TE): >? IGP
Choose margin type (absolute, relative, none): >? relative
Enter margin value: >? 2
Choose constraint type (affinity, disjoint, or 'done' to
finish): >? affinity
Choose affinity type (exclude-any, include-all, include-any):
>? exclude-any
Available affinity names:
[1] 100M
[2] 1G
[3] 10G
[4] 25G
[5] 40G
[6] 100G
[7] 100M+
[8] 1G+
[9] 10G+
[10] 25G+
[11] 40G+
```

```

[12] 100G+
[13] Maintenance
[14] Transnational
[15] Transcontinental
Enter affinity name numbers (comma-separated) for exclude-
any: >? 1,2,13
Choose constraint type (affinity, disjoint, or 'done' to
finish): >? disjoint
Choose disjoint type (link, node): >? link
Enter disjoint group ID: >? 1
Choose constraint type (affinity, disjoint, or 'done' to
finish): >? done
Enter preference for the path (or 'done' to finish): >? Done

```

Kod 4.1 Primjer ispisa skripte prilikom definiranja L2VPN P2P usluge

Nakon upisanih podataka za SR politiku u prvom smjeru, ako nema eksplicitno definiranih putanja unutar te konfiguracije, korisnika se pita želi li jednako konfiguriranu SR politiku u obrnutom smjeru, a u protivnom se cijeli postupak ponavlja još jednom. Razlog tome je što se unutar eksplicitne putanje mogu nalaziti SID-ovi susjedstva, koji ne moraju biti jednaki na oba usmjeritelja koji su povezani tom vezom na koju se oznaka odnosi. Jednako tako, ako je korisnik definirao `group-id` za disjunktnu putanju, prilikom (automatski odabranog) kreiranja SR politike za drugi smjer, korisnika se pita da upiše novi `group-id` kako se ne bi mrežni promet tih dvaju kreiranih SR politika između istih usmjeritelja izbjegavao kretati istim putanjama. Iz svih unosa korisnika generira se prilagođena struktura podataka koja opisuje obje SR politike. Na kraju se generira kompletan konfiguracijski kod koji uključuje i konfiguraciju L2VPN usluge na oba usmjeritelja, te se cijela struktura podataka zajedno sa konfiguracijskim linijama sprema u `sr_policies.json`.

Nakon kreiranja L2VPN usluge, korisnik ju može i izbrisati. Prilikom odabira opcije za brisanje, korisnika se traži unos IP adresa usmjeritelja na kojima želi izbrisati SR politike i L2VPN usluge. Skripta ispisuje sve trenutno konfigurirane SR politike na tim usmjeriteljima, a korisnik odabire koje od njih želi izbrisati. Njihovim odabirom skripta automatski briše i L2VPN uslugu koja koristi tu SR politiku za usmjeravanje prometa kroz mrežu. Primjer ispisa skripte kod brisanja jedne L2VPN P2P usluge može se vidjeti u Kod 4.2

```

.....
Enter IP address of the first router: 10.0.0.1
Enter IP address of the second router: 10.0.0.10

Xconnect services on 10.0.0.1:
[1] testXconnPolicy
Enter numbers of policies to delete (comma-separated): 1

Deleted policies on 10.0.0.1:
testXconnPolicy

Xconnect services on 10.0.0.10:
[1] testXconnPolicy
Enter numbers of policies to delete (comma-separated): 1

Deleted policies on 10.0.0.10:
testXconnPolicy

```

Kod 4.2 Primjer ispisa skripte prilikom brisanja L2VPN P2P usluge

4.3. Dohvaćanje topologije, računanje i primjenjivanje izračunatih TE metrika

Jedna od glavnih fokusa ovog rada je na osmišljavanju i implementaciji mehanizma koji će omogućiti SR politikama automatsko izbjegavanje zagušenja unutar mreže i iskorištavanje najoptimalnijih puteva u trenutnom vremenu. Za tu svrhu koristi se TE metrika unutar SR-a koja predstavlja metriku baziranu na trenutnoj propusnosti sučelja tj. veze. Ideja potječe iz *Enhanced Interior Gateway Routing Protocola* (EIGRP) koji kao jednu od komponenti metrike prema nekom IP prefiksu može koristiti opterećenje sučelja (engl. *load*). No problem s korištenjem EIGRP-a je u tome da EIGRP izračunava opterećenje sučelja prilikom učenja IP prefiksa. To znači da metrika prema nekom naučenom prefiksu može ostati ista jako dugo vremena, iako se stanje na sučeljima aktivno mijenja, što dovodi do toga da metrika nije odraz stvarnog stanja mreže [26].

U sklopu ovog rada, TE metrika jednog linka računa se po formuli (4)

$$TE_i = \frac{1}{1 - y_i} * IGP_i \quad (4)$$

TE_i je prirodni broj koji označava vrijednost TE metrike jedne veze, IGP_i je IGP metrika veze i , a y_i je vrijednost u rasponu $[0,1]$ koja označava opterećenje sučelja u izlaznom smjeru gdje 0 označava da kroz sučelje ne izlazi nikakav promet dok 1 označava maksimalnu iskorištenost sučelja uz izlaznom smjeru. Ako se dogodi da opterećenje y_i bude 1 onda se za vrijednost uzima maksimalna TE vrijednost koja se može konfigurirati (2147483647)[30]. Unutar Cisco usmjeritelja y_i se zapisuje u obliku zapisanom u formuli (5), gdje x_i je uvijek prirodni broj.

$$y_i = \frac{x_i}{255} \quad (5)$$

Kumulativna metrika neke SR politike jednaka je za sve vrste metrika, pa tako i za TE metriku te se računa prema formuli (6). Ako je suma TE vrijednosti veća od maksimalne koja se može konfigurirati na usmjeritelju, koristi se maksimalna konfigurabilna TE vrijednost.

$$TE_{total} = \sum_i TE_i \quad (6)$$

Prilikom odabira ove opcije unutar skripte, poziva se funkcija `get_te_metrics_and_rxload_and_srgb_map`. Ova funkcija sastoji se od dva dijela. Prvi dio sa svakog usmjeritelja dohvaća sve njemu poznate SID-ove unutar MPLS *forwarding* tablice i sučelja na koja se paketi prosljeđuju ako dođu na usmjeritelj sa tim SID-om. Uz svaki SID koji se nalazi u tablici (bio on globalni ili lokalni) sprema se koji je sljedeći usmjeritelj (u obliku globalnog SID-a) na koji će paket doći ako se pošalje na određeno sučelje, osim u slučaju kada je sljedeći usmjeritelj onaj na koji se osnovni SID odnosi. Cijela

struktura podataka koja opisuje topologiju unutar SR domene koja sadržava sve usmjeritelje i njima poznate SID-ove sa sljedećim skokovima sprema se u `router_srgbs.json` datoteku i može se vidjeti u isječku Kod 4.3 JSON datoteke gdje se prikazuju SID-ovi koje poznaje usmjeritelj sa oznakom (globalnog SID-a) 16001 i s *loopback* adresom 10.0.0.1.

```
{
  "16001": {
    "loopback_ip": "10.0.0.1",
    "sids": {
      "16002": "Gi0/0/0/0",
      "16003": "Gi0/0/0/1",
      "16004": [
        16002,
        "Gi0/0/0/0"
      ],
      "16005": [
        16002,
        "Gi0/0/0/0"
      ],
      "16006": [
        16003,
        "Gi0/0/0/1"
      ],
      , . . . . .
    }
  }
}
```

Kod 4.3 Isječak `router_srgbs.json` datoteke

Drugi dio funkcije dohvaća *Txload* (izlazno opterećenje) i trenutne IGP vrijednosti sa svih sučelja koja sudjeluju u SR-u i IGP susjedstvima i računa TE metriku prema formuli (4) na svim usmjeriteljima koji su dio SR domene, tj. usmjeriteljima koji su spojeni na PCE unutar skripte. Nakon izračunavanja trenutne TE metrike za svaku vezu, otvara datoteku `router_data.json` i čita prethodne TE metrike. Konačnu TE metriku izračunava koristeći eksponencijalno zaglađivanje s parametrom $\alpha = 0.7$ prema formuli (7). Ako JSON datoteka još ne postoji ili ne postoji vrijednost prethodno izračunate TE metrike za neko sučelje, skripta će primijeniti formulu (4). Iako se *txload* parametar na svakom sučelju mjeri kao prosjek u zadnjih pet minuta te se taj interval u kojem se računa prosjek može i mijenjati [27], kako bi se pri izračunu izbjegao utjecaj mogućih kratkotrajnih vršnih opterećenja sučelja ili općenito mogućih kratkotrajnih značajnih promjena na sučelju koje mogu biti

posljedice raznih drugih procesa (npr. kratkotrajni ispad veze) korišteno je eksponencijalno zaglađivanje čiji parametar ISP unutar svoje mreže može sam podesiti prema svojim zahtjevima, bez potrebe za mijenjanjem vremenskog intervala za računanje opterećenja sučelja na svim usmjeriteljima unutar mreže.

$$TE_{i,j} = \alpha * TE_{i,j} + (1 - \alpha) * TE_{i,j-1} \quad (7)$$

Drugi razlog korištenja eksponencijalnog zaglađivanja je granulacija promjena putanja SR politika koje koriste TE metriku. Pod pretpostavkom mogućeg velikog broja SR politika s aktivnom putanjom baziranoj na TE metrici, eksponencijalno zaglađivanje pomaže u sprječavanju mogućih naglih dodatnih promjena unutar mreže ISP-a uzrokovanih (mogućim) promjenama putanja SR politika.

Nakon izračuna novih TE vrijednosti za svako sučelje, skripta poziva funkciju `generate_and_apply_te_metric_config` koja te vrijednosti primjeni unutar konfiguracije svakog usmjeritelja i sprema ih u prije spomenutu datoteku `router_data.json` čiji se primjer može vidjeti u isječku json Kod 4.3. Struktura podataka datoteke je napravljena na način da se za svaki usmjeritelj spremaju informacije o *txload* parametru, IGP i TE metrici za svako sučelje.

```
{
  "10.0.0.1": {
    "GigabitEthernet0/0/0/0": {
      "igp_metric": 25,
      "txload": 0,
      "te_metric": 25
    },
    "GigabitEthernet0/0/0/1": {
      "igp_metric": 40,
      "txload": 0,
      "te_metric": 40
    },
  },
}
```

Kod 4.4 Isječak `router_data.json` datoteke

Skripta koristi json datoteke umjesto relacijske baze podataka iz razloga što se datoteke koriste više u svrhu privremene memorije te će funkcionalnost skripte ostati ista (osim u slučaju `sr_policies.json` datoteke), a omogućuje i jednostavan pregled podataka bez oslanjanja na druge tehnologije.

4.4. Izračun margina za SR politike s aktivnom dinamičnom putanjom s TE metrikom

Drugi dio mehanizma koji omogućava SR politikama automatsko izbjegavanje zagušenja unutar mreže i iskorištavanje najoptimalnijih puteva u trenutnom vremenu bazira se na dinamičnom izračunu margina prema karakteristikama putanje kojom prolaze i karakteristikama SR politike koja koristi TE metriku. Margina uz TE metriku dodatno omogućava ISP-u da upravlja putanjama SR politika. Korisnik odabirom ove opcije u skripti poziva funkciju `calculate_and_update_margins`. Na početku ove funkcije poziva se funkcija `get_te_metrics_and_rxload_and_srgb_map` koja ponovno izračunava najnoviju topologiju SR domene, kao i najnovije podatke o `txload` parametrima na sučeljima svih usmjeritelja. Funkcija učitava `sr_policies.json` datoteku i iterira po svakoj SR politici na svim usmjeriteljima. Za svaku SR politiku poziva se funkcija `check_active_te_path` koja iz usmjeritelja na kojem je konfigurirana ta SR politika, dohvaća trenutno aktivnu putanju. Ako je aktivna putanja dinamična s TE metrikom, funkcija `check_active_te_path` vraća preferenciju, SID listu i (totalnu) trenutnu metriku putanje kojom SR politika prolazi, u protivnom vraća prazne vrijednosti. Nakon poziva te funkcije, u `calculate_and_update_margins` funkciji unutar svake SR politike spremljene u json datoteci uvodi se pet varijabli koje su potrebne za dinamično izračunavanje margina:

1. `activeTEpreference`: ova varijabla služi za spremanje numeričke oznake preferencije aktivne dinamične putanje koja koristi TE metriku u prošlom pozivu funkcije. Ako je ova varijabla prazna, to označava da u prošlom pozivu funkcije SR politika još nije bila konfigurirana ili nije imala aktivnu dinamičnu putanju s TE metrikom. U normalnom slučaju se koristi kao provjera je li SR politika još uvijek na istoj konfiguraciji.
2. `baseSidList`: U ovu varijablu se sprema najmanja SID lista koja je bila aktivna u prethodnim pozivima funkcija.

3. `nextSidList`: Slično prethodnoj varijabli, u ovu varijablu se sprema SID lista koja je trenutno aktivna, ali je veličinom veća od SID liste spremljene u `baseSidList`.
4. `lastActiveTEmetric`: varijabla u koju se sprema totalna metrika putanje dobivena funkcijom `check_active_te_path` u prethodnom pozivu funkcije.
5. `currentMargin`: varijabla koja sprema prethodno izračunatu i konfiguriranu vrijednost margine.

Mehanizam odabira putanje na usmjeriteljima uvijek preferira manju SID listu na način da se preferira osnovna IGP metrika, tj. margine se uvijek odnose na putanju opisanu manjom SID listom. Npr. Za dvije putanje prema istom odredištu, prva sa veličinom SID liste 3 i druga s veličinom SID liste 4, konfigurirana margina će se uvijek odnositi na prvu SID listu. Ako je prvoj SID listi TE metrika 60, a drugoj SID listi TE metrika 50 i konfigurirana apsolutna margina 10, putanja SR politike će ići prvom SID listom i neće se dogoditi da druga (veća) SID lista ima veću TE metriku, a da bude aktivna u SR politici [16]. Zbog tog mehanizma, potrebno je prilagoditi izračun margine, kako se SR politika ne bi vraćala na putanju sa manjom SID listom, iako je to možda manje optimalna putanja.

Unutar skripte razrađeni su mogućnosti promjena putanja SR politika te se prema tim scenarijima računaju i konfiguriraju margine. Ispunjenje uvjeta za sljedeće mogućnosti provjeravaju se sekvencijalno, što znači iako bi se ispunjeni uvjeti mogli tumačiti višeznačno, prioritet imaju one mogućnosti koje su prve definirane.

Prvi i osnovna mogućnost je kada funkcija `check_active_te_path` vrati praznu varijablu `preference`. To označava da SR politika koja se trenutno pregledava nema aktivnu dinamičnu putanju s TE metrikom te se sve prije navedene varijable brišu i skripta kreće s procesiranjem sljedeće SR politike.

Sljedeća mogućnost je kada funkcija `check_active_te_path` vrati neku vrijednost u varijabli `preference`, ali ta vrijednost nije jednaka onoj spremljenoj u varijabli `activeTEpreference`. Ova situacija se može dogoditi iz razloga što do sada SR politika nije imala aktivnu dinamičnu putanju s TE metrikom ili se promijenila preferencija aktivne putanje. U oba slučaja potrebno je nanovo izračunati marginu za SR politiku koja se računa prema formuli (8)

$$Margin_i = \sum_j \frac{1}{1 - Y} * IGP_j - \frac{1}{1 - (y_j - Pol_{i,j})} * IGP_j \quad (8)$$

$Margin_i$ označava izračunatu vrijednost apsolutne margine za SR politiku i te se računa na način da se za svaku vezu j kojom prolazi SR politika, zbrajaju izrazi unutar sume. Ako je cijeli izraz (suma) negativan, uzima se vrijednost 0. Y je statički realni broj između 0 i 1 koji ISP može sam odrediti prema želji, a predstavlja maksimalnu željenu iskorištenost veza unutar mreže. U skripti se kao zadana vrijednost koristi 0.7 iz razloga što se 70% iskorištenosti veze generalno smatra kao gornja željena i sigurna granica [28][29]. Kao i u formuli (4), y_j predstavlja vrijednost u rasponu [0,1] koja označava opterećenje sučelja u izlaznom smjeru te se sa Cisco usmjeritelja dohvaća u obliku opisanom u formuli (5), dok $Pol_{i,j}$ označava dio opterećenja sučelja j koje je posljedica mrežnog prometa SR politike i te se računa po formuli (9)

$$Pol_{i,j} = \frac{S_{output_i} [Mbit/s] * IGP_j}{1000000 [Mbit/s]} \quad (9)$$

Pošto Cisco usmjeritelji informacije o izlaznoj brzini prijenosa podataka daju u mjernoj jedinici bitova po sekundi (bit/s), potrebno je konvertirati S_{output} varijablu dohvaćenu s usmjeritelja u mjernu jedinicu megabita po sekundi (Mbit/s) prije uvrštavanja u formulu (9).

Računajući marginu SR politike formulom (8) dobivamo vrijednost margine koja je ovisna o trenutnim (izlaznim) opterećenjima sučelja ali i o vlastitom udjelu opterećenja svakog pojedinog sučelja koje je posljedica prolaska SR politike tim sučeljem, tj. vezom. Tom se formulom omogućava da SR politike preferiraju putanje koje su najbolje i prema IGP metrici na način da će SR politike koje imaju veći udio prometa po vezama do odredišta ostajati duže na tim istim vezama od onih s manjim udjelom, jer je poželjno da SR politike koje rade veći mrežni promet prolaze vezama većih brzina prijenosa, tj. vezama većih IGP metrika kako ne bi došlo do promjene putanja i zagušenja veza s manjim brzinama prijenosa i prelaženja definirane gornje granice opterećenja.

Vraćajući se na prethodno opisanu mogućnost, nakon izračunavanja margine, unutar datoteke (u varijable `activeTEpreference`, `baseSidList` i `lastActiveTEmetric`) spremaju se podaci koje vrati funkcija `check_active_te_path`, a u varijablu `currentMargin` sprema se izračunata margina koja se ujedno konfigurira na SR politici. Varijabla `nextSidList` se postavlja na praznu vrijednost jer se radi o prvom računanju margine otkako je aktivna putanja s trenutnom preferencijom, te se trenutna putanja tj. trenutna SID lista smatra onom najmanjom za trenutno stanje mreže.

Sljedeća mogućnost je kad SID lista koju vrati funkcija `check_active_te_path` je veća od varijable `baseSidList` i varijable `nextSidList`, a varijabla `nextSidList` nije prazna. Ova mogućnost se odnosi na slučajeve kada se putanja SR politike promijeni po drugi ili n -ti ($n \geq 2$) put na SID listu koja je veća od SID lista spremljenih u `baseSidList` i `nextSidList` varijablama. Kada se to dogodi, to označava da se stanje u mreži na vezama (po kojima bi SR politika išla preferirajući IGP metriku) pogoršalo minimalno dvaput. U ovom slučaju varijabla `baseSidList` poprima vrijednost `nextSidList` varijable, dok `nextSidList` varijabla poprima vrijednost trenutne SID liste SR politike, tj. povrat funkcije `check_active_te_path`. Iz razloga preferencije IGP putanja zbog kraćih SID lista, vrijednost margine računa se po formuli (10). $Margin_n$ predstavlja novu marginu, dok $Margin_{n-1}$ predstavlja prošlo izračunatu marginu, a X predstavlja realni pozitivni broj u rasponu $[0,1]$ koji je unutar skripte postavljen kao 0.7, ali se može odrediti prema želji.

$$Margin_n = Margin_{n-1} * X \quad (10)$$

Postoje tri razloga ovakvog računanja margine u ovom slučaju. Prvi je zbog zagušenja mreže. Kako se osnovne veze kojom je SR politika prethodno prolazila zagušuju, želi se osigurati agresivnija potraga za boljim putanjama makar imale veću SID listu. Drugi razlog je sprječavanje stalnog mijenjanja putanja prilikom svakog izračuna TE metrika koji bi se mogli pojaviti ako je vrijednost (totalne) TE metrike putanje s manjom SID listom (tj. prošle putanje) zbrojena s marginom veoma blizu vrijednosti TE metrike s većom SID listom (tj. trenutne putanje), bilo zbog utjecaja SR politike ili drugih procesa. Iako se neće dogoditi

fizički gubitci paketa zbog načina na koji SR radi, stalno mijenjanje putanje uvodi dodatnu nestabilnost u mrežu ISP-a te može smetati protokolima poput TCP-a zbog promjene kašnjenja prema odredištu [31] uzrokovano promjenom putanje. Treći razlog je mogućnost vraćanja na osnovnu putanju (zapisane u `baseSidList` ili neku još manju SID listu) zbog izračuna margine po formuli (8) koja bi na putanji s većom SID listom zbog veće totalne IGP metrike vrlo moguće vratila veću vrijednost margine nego je bila prethodno definirana, što bi posljedično moglo uzorkovati vraćanje SR politike na putanju (nanovo) zapisanu u `baseSidList` varijablu (ili još više zagušenu putanju opisanu još manjom SID listom) koju je SR politika napustila prilikom zadnjeg izračuna i konfiguriranja TE metrika. Drugim riječima, formulom (10) se osigurava da se SR politika ne smije vratiti na putanju s manjom SID listom ako se njezina totalna TE metrika nije poboljšala barem za 30% (ako je $\times 0.7$) prijašnje vrijednosti margine

Osim izračunate margine i novih SID lista, sprema se i trenutna totalna TE metrika SR politike u varijablu `lastActiveTEmetric` te su konfiguraciji usmjerenja primjenjuje izračunata margina.

Sljedeća mogućnost koja se provjerava je ako je varijabla `nextSidList` jednaka trenutnoj SID listi koju vrati funkcija `check_active_te_path`. Ako jest, mijenja se samo varijabla `lastActiveTEmetric` da prikazuje osvježeno stanje totalne TE metrike.

Mogućnost slična prethodnoj je kada je veličina SID liste spremljene u `nextSidList` varijabli jednaka veličini SID liste koju vrati funkcija `check_active_te_path`, ali su te dvije liste drugačije. Ova situacija je moguća ako se promijenila putanja SR politike, ali na putanju koja je reprezentirana jednakom duljinom SID liste. U ovom slučaju u varijablu `nextSidList` sprema se povrat funkcije `check_active_te_path`, ažurira se `lastActiveTEmetric` varijabla, a margina se računa, pa potom sprema i konfigurira prema formuli (10) iz istih razloga navedenih ranije.

Sljedeća mogućnost po redu se odnosi na slučaj kada funkcija `check_active_te_path` vrati veličinom manju ili jednaku SID listu od one spremljene u varijabli `baseSidList`. Ovaj uvjet opisuje situaciju kada SR politika ostane na postojećoj poznatoj osnovnoj putanji s najmanjom SID listom, ili se prebaci na putanju sa SID listom još manjom od one spremljene u `baseSidList` varijabli. Skripta u tom slučaju radi sve isto kao i u prvoj mogućnosti, tj. u variable `baseSidList` i `lastActiveTEmetric` sprema povratne

varijable funkcije `check_active_te_path`, varijablu `nextSidList` se postavlja na praznu vrijednost, a marginu računa, sprema i konfigurira prema formuli (8)

Predzadnja mogućnost odnosi se na slučaj kada SID lista koju vrati funkcija `check_active_te_path` je veličinom veća od SID liste zapisane u varijabli `baseSidList`, a `nextSidList` je prethodno postavljen na praznu vrijednost. Ovaj slučaj označava prijelaz SR politike s osnovne putanje zapisanoj u `baseSidList` varijabli, na neku drugu putanju s većom SID listom zbog nepovoljnijih TE metrika na osnovnoj putanji. Slično kao i u prethodnim slučajevima gdje SR politika prelazi na veću SID listu, margina se računa i konfigurira po formuli (10), a u varijable `nextSidList` i `lastActiveTEmetric` se spremaju povratu funkcije `check_active_te_path`.

Zadnja mogućnost odnosi se na slučaj sličan prethodnom, ali uz uvjete da SID lista koju vrati funkcija `check_active_te_path` je veća od `baseSidList`, ali manja od `nextSidList` varijable. Ovi ispunjeni uvjeti označuju promjenu TE metrika na način da se najbolja putanja približava osnovnoj putanji s najmanjom SID listom na neku putanju veličinom SID liste između prijašnje trenutne putanje zapisane u `nextSidList` varijabli i osnovne putanje `baseSidList`. U ovom slučaju potrebno je izračunati i konfigurirati marginu po formuli (8), u varijable `baseSidList` i `lastActiveTEmetric` spremiti povrat funkcije `check_active_te_path`, a varijablu `nextSidList` postaviti na praznu vrijednost. Moguće je da će se putanja SR politike prilikom primjenjivanja margine vratiti na putanju s još manjom SID listom zbog razloga navedenih kod formule (8) iako to nije trenutno najbolja putanja, ali će se već sljedećom iteracijom računanja margina primijeniti neka od prošlo opisanih mogućnosti koje SR politiku vraćaju na trenutno najbolju putanju.

4.5. Automatski izračuni TE metrike i margine

Zadnja opcija koja se prezentira korisniku skripte je pokretanje automatskog pozadinskog procesa koji periodično izvršava sljedeće:

1. poziva funkcije `get_te_metrics_and_rxload_and_srgb_map` i `generate_and_apply_te_metric_config`
2. Čeka 5 minuta
3. Poziva funkciju `calculate_and_update_margins`

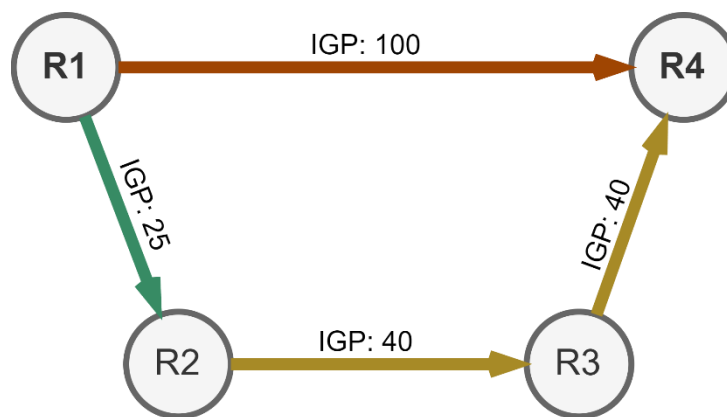
4. Čeka 5 minuta

Time se korisniku skripte omogućuje da ima periodično izračunavanje i konfiguriranje TE metrika svih veza unutar mreže, kao i svih margina SR politika koje imaju trenutno aktivnu dinamičnu putanju baziranu na TE metrici. Drugim riječima, omogućuje se kompletno automatizirano upravljanje dijelova mreže koji su bitni za uspostavu primarno L2VPN, ali i drugih usluga koje ISP želi koristiti oslanjajući se na SR-TE. Razlog čekanja između računanja TE metrika i margina je zadan interval mjerenja prosjeka *txload* parametra na sučeljima usmjerenja opisanog u sekciji 4.3. Korisnik u svakom trenutku unutar CLI-a može prekinuti izvršavanje pozadinskog procesa pritiskom tipke *q* na tipkovnici.

4.6. Nedostaci skripte u računanju TE metrika i margina

Načini izračunavanja TE metrika i margina opisanih u ovom radu ima nekoliko vidljivih nedostataka od kojih se neki mogu relativno lako izbjeći, a drugi predstavljaju logički problem koji ISP ovisno o strukturi mreže treba riješiti na svoj način.

Prvi takav problem javlja se kada postoje dvije putanje unutar mreže sličnih IGP metrika ilustrirano na Slika 4.1



Slika 4.1 Primjer mreže s dvije putanje sličnih IGP metrika

Za neku uslugu koja koristi 6gbit/s prometa, ide od usmjerenja 1 do usmjerenja 4 i unutar SR politike koristi se TE metrika. S pretpostavkom da su sve veze prazne i da su im TE metrike iste kao i IGP metrike, SR politika će krenuti direktnom putanjom prema usmjerenju 4. No ako se pretpostavi da je ta direktna putanja opterećena s 10% svojih kapaciteta, tada prilikom svake iteracije izračuna TE metrika i margina prema funkcijama u

skripti SR politika će mijenjati svoju putanju što dovodi do neželjenih fluktuacija u mreži. Drugi, sličan primjer, odnosi se na usluge koje nadmašuju željeno opterećenje najbolje putanje. Pod pretpostavkom da usluga koristi 9gb/s prometa, jednako kao i u prošlom primjeru, prilikom svake iteracije računanja TE metrika i margina, SR politika će mijenjati svoju putanju bez ikakvih utjecaja vanjskih varijabli.

Za oba navedena slučaja bilo bi poželjno, ako količina prometa usluge predstavlja veći dio opterećenja nekog dijela putanje i ako je stalna, da se koristi IGP metrika, a kritična veza kojom prolazi SR politika poveća kapacitetom po potrebi. Druga mogućnost je korištenje mape afiniteta kako bi se osiguralo da SR politika prolazi vezama zadovoljavajućeg kapaciteta na koje neće imati značajan utjecaj, što bi u ovom slučaju bilo *include-any 25G+*. Potrebno je napomenuti kako sa korištenjem mape afiniteta treba biti oprezan, pošto se lako može dogoditi da cijela SR politika dođe u *invalid* stanje jer ne može ispuniti sve zadane uvjete, ili da promet krene okolnim putanjama koji značajno povećavaju kašnjenje i iskorištavanje resursa.

Treći primjer nedostatka skripte odnosi se na zagušenje jedne veze unutar putanje. Moguća je situacija u kojoj je jedna veza unutar putanje SR politike zagušena, ali zbog strukture mreže i načina računanja (tj. zbrajanja pojedinačnih) TE metrika neće promijeniti putanju. Iako je ovakav slučaj malo vjerojatan zbog toga što TE metrika svake veze za 70% opterećenja postaje 3.33 puta veća, a za 90% opterećenja postaje čak 10 puta veća od svoje IGP metrike, usluga bi bila izrazito loše kvalitete. Jedna od mogućih solucija je mijenjanje logike izračunavanja margina na način da se za svaku politiku, neovisno o SID listi, provjerava ako je jedna od veza kojom prolazi SR politika opterećena preko 70% (ili postotka koji odredi ISP) bez opterećenja koje radi SR politika. Ako postoji takva veza, neovisno o trenutno konfiguriranoj margini, margina SR politike se konfigurira na 0. Tim se omogućuje agresivnija potraga za boljom putanjom, no uvodi problem mogućih fluktuacija. Jedino drugo rješenje, ako ne postoji alternativna putanja, je povećavanje kapaciteta problematične veze.

Zaključak

U ovom radu istražena je primjena SR-TE-a za optimizaciju mrežne infrastrukture ISP-a. Rad se usredotočio na pružanje L2VPN P2P usluga, ističući izazove skalabilnosti i složenosti konfiguracije s kojima se suočavaju tradicionalni MPLS-TE pristupi. SR se pojavljuje kao rješenje koje pojednostavljuje upravljanje prometom i omogućuje fleksibilnije usmjeravanje.

Rad detaljno opisuje osnovne koncepte MPLS-a i MPLS-TE-a, naglašavajući ulogu labela, LSP-ova, LDP i RSVP-TE protokola. Također je predstavio SR kao modernu alternativu koja koristi segmente tj. SID-ove za definiranje putanja prometa. SR politike omogućuju preciznu kontrolu nad prometom, što je posebno korisno za usluge poput L2VPN P2P usluga.

Rad je također pokazao da SR, u kombinaciji s TE, nudi značajne prednosti u odnosu na tradicionalne MPLS-TE pristupe. SR omogućuje jednostavniju konfiguraciju, bolju skalabilnost i fleksibilnije usmjeravanje prometa. Korištenje dinamičkih putanja i PCE-a dodatno poboljšava prilagodljivost mreže promjenjivim uvjetima prometa. Rad također ukazuje na izazove i ograničenja primjene SR-TE-a, pogotovo pri izboru i izračunu odgovarajućih metrika i margina za optimizaciju putanja.

Ključni doprinos ovog rada je razvoj Python skripte koja automatizira kreiranje L2VPN P2P usluga temeljenih na SR politikama. Skripta pojednostavljuje konfiguraciju mrežnih uređaja omogućujući jednostavno definiranje putanja, metrika i ograničenja. Skripta omogućuje automatizaciju izračuna TE metrika i margina na inovativan način te optimizaciju mrežnog prometa baziranog na SR-TE-u s TE metrikom. Korištenjem GNS3 simulatora, rad je uspješno simulirao realistično mrežno okruženje s Cisco IOS XRv9000 usmjeriteljima, potvrđujući valjanost predloženih rješenja.

Daljnja istraživanja mogla bi se usmjeriti na istraživanje primjene SR-a u drugim vrstama mrežnih usluga poput BPG *peeringa* te testiranje ponuđenog rješenja unutar realne mreže ISP-a.

Literatura

- [1] Sturt R., Rosencrance L., Scarpatti J., *Multiprotocol Label Switching*, TechTarget, (2023, ožujak). Poveznica: <https://www.techtarget.com/searchnetworking/definition/Multiprotocol-Label-Switching-MPLS>, pristupljeno 23. svibnja 2024.
- [2] Sužnjević M., *Mrežni promet i performanse*, FER, (2022, listopad), Poveznica: <https://www.fer.unizg.hr/download/repository/MPP%201%20-%20Arhitektura%20i%20tehnologije%20dana%20C5%A1njih%20mre%20C5%BEa.pdf>, pristupljeno 27. svibnja 2024.
- [3] Hewlett Packard Enterprise Development LP, *MPLS Forwarding*, (2016.), Poveznica: https://techhub.hpe.com/eginfolib/networking/docs/switches/5820x-5800/5998-7393r_mpls_cg/content/441756023.htm, pristupljeno 27. svibnja 2024.
- [4] Brittain, P., & Farrel, A. (2000). *MPLS traffic engineering: A choice of signaling protocols*. Data Connection, (2000), str. 10
- [5] Garcia J., *LDP vs RSVP*, CCDE Certification Community, (2020, veljača). Poveznica: <https://learningnetwork.cisco.com/s/question/0D53i00000KsqPCCAZ/ldp-vs-rsvp>, pristupljeno 27. Svibnja 2024
- [6] DiVincenzo M., *Segment Routing: The Future of MPLS*, (2023, svibanj), Poveznica: <https://www.wwt.com/article/segment-routing-the-future-of-mpls>, pristupljeno 27. svibnja 2024
- [7] Michielsen K., Filsfils C. *Segment Routing Introduction*, Segment-routing, (2015, studeni), Poveznica: <https://www.segment-routing.net/tutorials/2016-09-27-segment-routing-introduction/>, pristupljeno: 27. svibnja 2024.
- [8] Michielsen K., Filsfils C. *SR Traffic-Engineering*, (2015, studeni), Poveznica: <https://www.segment-routing.net/tutorials/2017-03-06-segment-routing-traffic-engineering-srte/>, pristupljeno: 27. svibnja 2024.
- [9] Pereira, M., *Traffic Engineering based on MPLS and Segment Routing*, Diplomski rad, Departamento de engenharia electrotécnica e de computadores, 2021.
- [10] Dooley K., *Finding & Fixing Asymmetric Routing Issues*, Poveznica: <https://www.auvik.com/franklyit/blog/asymmetric-routing-issue/>, pristupljeno: 31. svibnja 2024.
- [11] Cisco, *MPLS Traffic Engineering Path Link and Node Protection Configuration Guide, Cisco IOS XE Gibraltar 16.11.x*, (2019, lipanj), Poveznica: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mp_te_path_protect/configuration/xe-16-11/mp-te-path-protect-xe-16-11-book/mpls-traffic-engineering-fast-reroute-link-and-node-protection.html, pristupljeno: 31. svibnja 2024.
- [12] Yasini, S. M. J. *Implementation of segment routing and MPLS traffic engineering in software-defined network based on GNS3 network emulator and OpenDayLight SDN controller*, Diplomski rad, Dipartimento di Elettronica, Informazione e Bioingegneria, 2017.

- [13] Frost D., Bryant S., RFC 6374: Packet Loss and Delay Measurement for MPLS Networks, 2011.
- [14] Filisfilis C. *SR MPLS – Performance Monitoring*, (2018, studeni), Poveznica: <https://www.segment-routing.net/tutorials/2018-03-06-sr-delay-measurement/>, Pristupljeno: 31. svibnja 2024.
- [15] Nokia, *TE Metric and IGP Metric*, (2023.), Poveznica: <https://infocenter.nokia.com/public/7705SAR234R1A/index.jsp?topic=%2Fcom.nokia.mpls-guide%2Fte-metric-igp-metric.html>, Pristupljeno: 1. lipnja 2024.
- [16] Cisco, *Cisco WAE Design 6.3.1 User Guide*, (2016. ožujak), Poveznica: https://www.cisco.com/c/en/us/td/docs/net_mgmt/wae/6-3-1/design/user/guide/WAE_Design_User_Guide.html, Pristupljeno: 3. lipnja 2024
- [17] Ohanian A., *The (Unofficial) CCNP-SP Study Guide – SR-TE Pt. 3*, (2022. prosinac), Poveznica: <https://ccnp-sp.gitbook.io/studyguide/mps/segment-routing/sr-te-pt.-3-using-a-pce>, Pristupljeno: 3. lipnja 2024.
- [18] Cisco, *Segment Routing Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.5.x*, (2022. travanj), Poveznica: <https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/75x/b-segment-routing-cg-ncs5500-75x/configure-sr-te-policies.html>, Pristupljeno: 3. lipnja 2024.
- [19] Vasseur J. P., Le Roux J. L. RFC 5440: *Path computation element (PCE) communication protocol (PCEP)*, 2009.
- [20] Kalmykov N. S., Dokuchaev V. A., *Segment routing as a basis for software defined network*. Т-Comm-Телекоммуникации и Транспорт, 15,7 (2021), str. 50-54.
- [21] Singh D., *PCE and PCEP Overview*, (2015. listopad), Poveznica: <https://packetpushers.net/blog/pce-pcep-overview/>, Pristupljeno: 3. lipnja 2024.
- [22] Grossmann J., Yuen S., *What is GNS3*, (2020, srpanj), Poveznica: <https://github.com/mother/gns3-docs/blob/master/docs/getting-started/what-is-gns3.md>, Pristupljeno: 3. lipnja 2024.
- [23] Ritter H., *IOS XRv 9000 - PCE/PCC Policy Down*, (2021. veljača), Poveznica: <https://community.cisco.com/t5/mps/ios-xrv-9000-pce-pcc-policy-down/td-p/4298043>, Pristupljeno: 6. lipnja 2024.
- [24] Cisco, *IP Routing: ISIS Configuration Guide, Cisco IOS XE Release 3S*, (2014. rujan), Poveznica: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_isis/configuration/xe-3s/irs-xe-3s-book/irs-fscbp.html, Pristupljeno: 6. lipnja 2024.
- [25] Cisco, *ASR9000/XR Introduction to Segment Routing*, (2017. svibanj), Poveznica: <https://community.cisco.com/t5/service-providers-knowledge-base/asr9000-xr-introduction-to-segment-routing/ta-p/3166037>, Pristupljeno: 6. lipnja 2024.
- [26] Harmoush E., *EIGRP Metric*, (2016. travanj), Poveznica: <https://www.practicalnetworking.net/stand-alone/eigrp-metric/>, Pristupljeno: 14. lipnja 2024.
- [27] Haubein T. *TX & RX*, (2016. travanj), Poveznica: <https://www.thepacket.net/load/>, Pristupljeno: 14. lipnja 2024

- [28] Solarwinds, *NetFlow Traffic Analyzer – getting started guide*, (2024, lipanj), Poveznica: <https://documentation.solarwinds.com/archive/pdf/nta/netflowgettingstartedguide.pdf>, Pristupljeno: 16. lipnja 2024.
- [29] Lamberti A., *Network Utilization Monitoring: How to Check Network Usage*, (2024, listopad), Poveznica: <https://obkio.com/blog/network-utilization-monitoring/>, Pristupljeno: 16. lipnja 2024.
- [30] Cisco, *Segment Routing Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.6.x*, (2022. ožujak), Poveznica: <https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/76x/b-segment-routing-cg-ncs5500-76x/configure-sr-te-policies.html>, Pristupljeno: 16. lipnja 2024.
- [31] Andreadis A., Rizzuto S., Zambon R. (2016). *A cross-layer jitter-based TCP for wireless networks*, Eurasip Journal on Wireless Communications and Networking, 2016, 1 (2016), str. 191
- [32] Cisco, *IOS XRv 9000 Router Release 7.7.2 MD*, (2022, listopad), Poveznica: <https://software.cisco.com/download/home/286288939/type/280805694/release/7.7.2>, Pristupljeno: 4. travnja 2024.
- [33] Duponchelle J., *Cisco IOS XRv 9000 appliance*, (2016, studeni), Poveznica: <https://www.gns3.com/marketplace/appliances/cisco-ios-xrv-9000>, Pristupljeno: 4. travnja 2024.

Sažetak

Razvoj upravljačke komponente za ostvarenje zahtjeva usluga unutar mreže pružatelja internetskih usluga

Ovaj diplomski rad istražuje primjenu *Segment Routinga* (SR) i *Traffic Engineeringa* (TE) za optimizaciju mrežne infrastrukture pružatelja internetskih usluga (ISP-a) s naglaskom na pružanje L2VPN P2P usluga. Rad opisuje osnovne koncepte MPLS-a i MPLS-TE-a, a zatim se usredotočuje na SR kao modernu alternativu. Ključni doprinos je razvoj Python skripte koja automatizira kreiranje L2VPN P2P usluga temeljenih na SR politikama te optimizaciju njihovih putanja korištenjem TE metrika i margina. Korištenjem GNS3 simulatora, za potrebe rada simulira se realistično mrežno okruženje s Cisco IOS XRv9000 usmjeriteljima.

Ključne riječi: Segment Routing (SR), Traffic Engineering (TE), dinamična putanja, metrika, margine, python, skripta

Summary

Development of a control component for fulfilling service requirements within an Internet service provider network

This thesis explores the application of Segment Routing (SR) and Traffic Engineering (TE) to optimize the network infrastructure of Internet Service Providers (ISPs) with an emphasis on providing L2VPN P2P services. The paper describes the basic concepts of MPLS and MPLS-TE, then focuses on SR as a modern alternative. The key contribution is the development of a Python script that automates the creation of L2VPN P2P services based on SR policies and optimizes their paths using TE metrics and margins. Using the GNS3 simulator, a realistic network environment with Cisco IOS XRv9000 routers is simulated for the purposes of this study.

Keywords: Segment Routing (SR), Traffic Engineering (TE), dynamic path, metrics, margins, python, script

Privitak

Za potrebe simulacije koja se koristila u izradi ovog rada, korišten je Dell PowerEdge R820 server sa sljedećim specifikacijama:

- 4x Intel Xeon E5-4620
- 1536GB DDR3 ECC 1333MHz RAM
- 2x Dell SATA SSD 960GB RAID1

Pokrenuta simulacija opisana u poglavlju 3 uzorkuje prosječno opterećenje procesora od 70% i korištenja 350GB RAM-a. Potreban prostor za spremanje svakog od instaliranih *firmwarea* usmjeritelja je nekoliko gigabajta po usmjeritelju.

Za korištenje skripte izrađene u sklopu ovog rada osim paketa koji su dio standardne Python biblioteke (`os`, `time`, `re`, `copy`, `json` i `threading` paketi) potrebno je instalirati dodatne pakete `tqdm`, `paramiko` i `keyboard` naredbom `pip install tqdm paramiko keyboard`