UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Koščak

Analiza upravljanja s podatkovnimi tokovi v računalniških omrežjih

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Miha Mraz

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01635/2010
Datum: 15.02.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGOR KOŠČAK**

Naslov: **ANALIZA UPRAVLJANJA S PODATKOVNIMI TOKOVI V RAČUNALNIŠKIH OMREŽJIH**

TRAFFIC FLOW MANAGEMENT ANALYSIS IN COMPUTER DATA NETWORKS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kandidat naj v svojem delu opiše osnovne pristope pri optimizaciji računalniških podatkovnih omrežij. Pri tem naj upošteva kriterija dolžinske metrike in možnosti tuneliranja. V svojem delu naj kandidat implementira zglede obeh konceptov in primerja njune rezultate.

Mentor:

prof. dr. Miha Mraz

Dekan:

prof. dr. Franc Solina

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01635/2010
Datum: 15.02.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGOR KOŠČAK**

Naslov: **ANALIZA UPRAVLJANJA S PODATKOVNIMI TOKOVI V RAČUNALNIŠKIH OMREŽJIH**

TRAFFIC FLOW MANAGEMENT ANALYSIS IN COMPUTER DATA NETWORKS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kandidat naj v svojem delu opiše osnovne pristope pri optimizaciji računalniških podatkovnih omrežij. Pri tem naj upošteva kriterija dolžinske metrike in možnosti tuneliranja. V svojem delu naj kandidat implementira zglede obeh konceptov in primerja njune rezultate.

Mentor:

prof. dr. Miha Mraz

Dekan:

prof. dr. Franc Solina

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Zahvala

Zahvaljujem se mentorju prof. dr. Mihi Mrazu za pomoč in nasvete pri nastajanju diplomskega dela.

Zahvaljujem se tudi mami za podporo v celotnem času študija ter Jerneji za vzpodbudne besede ob pisanju diplome in za jezikovni pregled besedila.

Kazalo

Seznam uporabljenih kratic in simbolov

BGP - Border Gateway Protocol
EGP - Exterior Gateway Protocol
EIGRP - Enhanced Interior Gateway Routing Protocol
GRE - Generic Routing Encapsulation
IGP - Interior Gateway Protocol
IP - Internet Protocol
ISIS - Intermediate System to Intermediate System
LDP - Label Distribution Protocol
LSP - Label Switched Path
MPLS - Multi Protocol Label Switching
OSPF - Open Shortest Path First
P - Provider
PE - Provider Edge
RIP - Routing Information Protocol
RSVP - Resource Reservation Protocol
TCP - Transmission Control Protocol
TDP - Tag Distribution Protocol
TE - Traffic Engineering
TFTP - Trivial File Transfer Protocol
TTL - Time To Live
UDP - User Datagram Protocol

Povzetek

Danes se podatkovna računalniska omrežja uporabljajo vsepovsod okoli nas. Večina teh omrežij zaradi zanesljivosti uporablja dodatne nadomestne povezave, ki pa so pogosto neizkoriščene. Razlog je v osnovni značilnosti usmerjevalnih protokolov, ki se uporabljajo za zaznavanje izpadov v omrežju, da vedno ves promet usmerijo po najboljši poti. S tem obremenijo optimalno pot, ostale pa so neuporabljene. Usmerjevalniki, ki se uporabljajo v bolj zahtevnih okoljih, poznajo mehanizme, s katerimi lahko del prometa usmerimo po poljubni poti, hkrati pa obdržimo sposobnost preusmeritve v primeru odpovedi dela omrežja. Ti mehanizmi se imenujejo prometno načrtovanje (angl. *traffic engineering*) in omogočajo izgradnjo navideznih povezav glede na omejitve omrežja in zahteve po kapacitetah. Diplomska naloga se ukvarja s primerjavo enostavnih pristopov, ki bi zagotovili enako funkcionalnost ali vsaj zadostili osnovnim potrebam po preusmeritvi dela prometa z odpornostjo na odpovedi. Opisani so pristopi z vplivanjem na metriko, uporaba deljenja prometnih tokov na več povezav in uporaba navideznih tunelskih povezav. Narejena je primerjava s prometnim načrtovanjem, analiza skalabilnosti in kompleksnosti implementacije.

Ključne besede:

MPLS, prometno načrtovanje, podatkovna omrežja, tok prometa, optimizacija

Abstract

Today, computer data networks are being used everywhere around us. To increase reliability most of such networks use redundant links, which may not be fully utilized at all times. The reason lies with routing protocols, which are used to detect network faults, and at all times direct all traffic only across the best path. By doing this only the optimal path is taken advantage of, while others are idle. High end routers have the capability to redirect part of the traffic across arbitrary path, at administrator's discretion, while maintaining fault tolerance. These mechanisms are referred to as traffic engineering. They allow to define virtual pathways which are bounded by capacity constraints of the network and demand of the traffic. This thesis compares different simple approaches to achieving the same functionality or at least allowing partial traffic redirection with fault tolerance. It describes approaches with metric modification, traffic load sharing, and use of traffic tunnels and compares these to traffic engineering regarding scalability and configuration complexity.

Keywords:

MPLS, traffic engineering, data networks, traffic flow, optimization

Poglavje 1

Uvod

Računalniška podatkovna omrežja imajo osnovno nalogo zagotavljanja povezljivosti med končnimi napravami. V najpreprostejši obliki je omrežje le ena povezava med dvema napravama. Ko število naprav narašča, število povezav raste. Z večanjem števila povezav odpravljamo dve vrsti nevšečnosti, ki se lahko pojavijo v omrežju, in sicer:

- prekinitev povezave in
- preobremenitev povezave.

Če hočemo, da prekinitev povezave v omrežju ne vpliva na povezljivost, se poslužimo redundance, kar pomeni, da dodamo novo povezavo. Če je naše omrežje veliko in povezave niso popolnoma pod našim nadzorom, potem vedno obstaja možnost, da pride do odpovedi, ne le zaradi naravnih vzrokov, temveč tudi zaradi človeškega faktorja, ki je izven naše kontrole. Podvajanje oziroma multipliciranje povezav je v splošnem edini način, da zagotovimo večjo zanesljivost omrežja.

Če omrežje gradimo v obliki drevesa, potem so veje, kjer se promet združuje, obremenjene bolj kot veje, bližje listom. Zato morajo imeti večjo prepustnost, analogno živim drevesom, kjer so veje bližje deblu debelejše. Če v omrežje vpeljemo nove povezave, lahko dosežemo, da se promet preusmeri v te povezave, namesto da obremeni centralne povezave.

V vseh omrežjih, ki imajo redundančne poti, potrebujemo mehanizme, ki usmerjajo promet v "najboljše" povezave, preprečujejo zanke, kamor bi se promet lahko ujel (in nikoli prišel na destinacijo) in se dinamično prilagajajo spremembam v omrežju. Ti mehanizmi se imenujejo usmerjevalni protokoli in predstavljajo način komunikacije med usmerjevalniki – napravami, ki usmerjajo promet (se odločajo, v katero smer bodo poslali promet). Usmerjevalne

protokole delimo glede na namembnost na notranje usmerjevalne protokole (IGP - Interior Gateway Protocol) ter zunanje usmerjevalne protokole (EGP - Exterior Gateway Protocol). Notranji usmerjevalni protokoli se nadalje delijo glede na način delovanja na dve skupini – t. i. "distance vector" in t. i. "link-state" skupino. Značilnost prve skupine je, da usmerjevalnik vidi sliko neposredne okolice, za ostale informacije pa zaupa izračunom svojih sosedov, medtem ko si druga skupina izdela pregled omrežja ter individualno izračuna optimalne poti. V obeh primerih algoritem usmerjevalnega protokola izbere najboljšo izhodno pot do podomrežja, v katerega so podatki namenjeni. Na ta način ne moremo polno izkoristiti redundančnih poti, ki nam tako služijo izključno kot nadomestne poti in so, če omrežje deluje, kot mora, neizkoriščene.

Cilj diplomske naloge je analiza različnih načinov nastavitev usmerjevalnikov, ki nam omogočajo, da v najboljši meri izkoristimo alternativne poti do destinacije, pri čemer pa moramo obvezno obdržati osnovno funkcionalnost dinamičnih usmerjevalnih protokolov in to je prilagodljivost na spremembe v omrežju. Prednosti večje izkoriščenosti povezav se pokažejo, ko začne promet po optimalnih poteh naraščati. Z optimizacijo lahko pri ustrezni topologiji omrežja povečamo izkoristek, zmanjšamo obremenjenost povezav, izgube paketov in zakasnitve na povezavah ter se na ta način izognemo dodatnim stroškom, ki bi nastali z nadgrajevanjem omrežja. V drugem poglavju diplomske naloge so opisani različni pristopi k optimizaciji prometnih tokov v podatkovnem omrežju z uporabo klasičnih metod manipulacije toka prometa. V tretjem poglavju je opisana rešitev z uporabo prometnega načrtovanja. Četrto poglavje vsebuje primerjavo dveh pristopov na konkretnem primeru omrežja.

Poglavje 2

Načini pristopa k optimizaciji omrežja

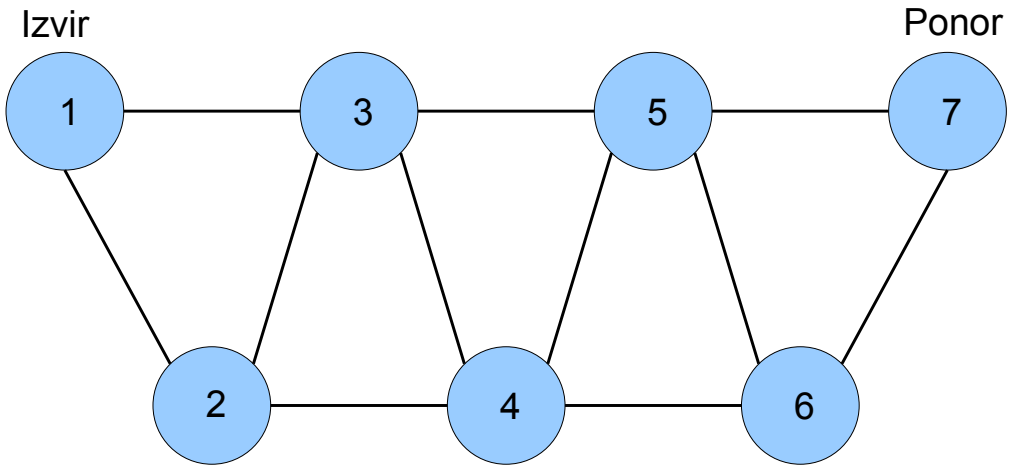
V tem poglavju so predstavljene različne metode, s katerimi lahko vplivamo na potek prometa skozi omrežje. Večina ima vsaj nekaj pomanjkljivosti, kar je podrobno razloženo v posameznih podpoglavjih.

V prvih dveh podpoglavjih sta opisana načina, s katerima lahko speljemo promet po različnih poteh in ne po tisti, po kateri bi se sicer usmeril. To lahko naredimo selektivno, tako da le del prometa usmerimo po novi poti, ostali promet pa se še vedno drži prvotne poti. Namen tega je, da delno razbremenimo primarne povezave, obremenimo pa tiste, ki so bile do zdaj neizkoriščene. Tretje podpoglavje nam opisuje, kako lahko promet do istega cilja speljemo po različnih poteh. Četrto podpoglavje pregleda postopke, s katerimi lahko promet preusmerimo glede na izvorni, in ne ciljni, naslov.

Strnjeno so ti pristopi naslednji:

- izbira alternativne poti z manipulacijo metrike,
- uporaba tunelskih povezav namesto manipulacije metrike,
- deljenje prometa:
  - deljenje prometa po paketih,
  - deljenje prometa pri različnih metrikah s protokolom EIGRP;
- usmerjanje prometa glede na izvorni naslov z usmerjanjem s pravili.

2.1 Izbira alternativne poti z manipulacijo metrike



Slika 2.1: Topologija testnega omrežja.

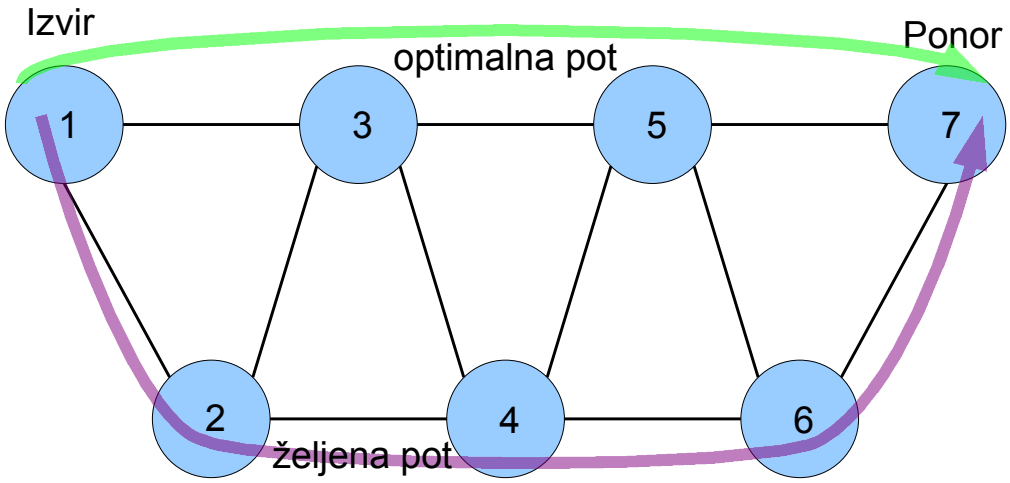
Uporabljen usmerjevalni protokol je RIP. Ta je bil izbran zaradi enostavnosti. Vsi ostali usmerjevalni protokoli tipa "distance vector" so prevedljivi na RIP, le metrika je bolj kompleksna. Pri RIP-u je metrika število usmerjevalnikov (skokov) do ciljnega omrežja [6]. Na sliki 2.1 vidimo topologijo testnega omrežja. Radi bi poslali promet od 1 k 7. Optimalna pot (in edina izbrana) je $1-3-5-7$.

Če želimo promet preusmeriti na drugo pot, potem moramo ustrezno utežiti neželjene povezave. Privzeta metrika oziroma cena vsake povezave je 1. Ker je cena poti število povezav, povezavo naredimo manj ugodno tako, da spremenimo povečanje metrike z 1 na > 1.

2.1.1 Prva naloga

V prvem primeru želimo promet preusmeriti na pot $1-2-4-6-7$, kot je predstavljeno na sliki 2.2. Neugodne povezave so $1-3$, $2-3$, $3-5$, $4-5$ in $5-7$. Povezavi $3-4$ in $5-6$ sta povratni in v vsakem primeru manj ugodni.

Na povezavah $1-3$, $2-3$, $3-5$ in $4-5$ nastavimo metriko 2. Na povezavi $5-7$ moramo metriko nastaviti na 3. Razlog za to je, da usmerjevalnik 5

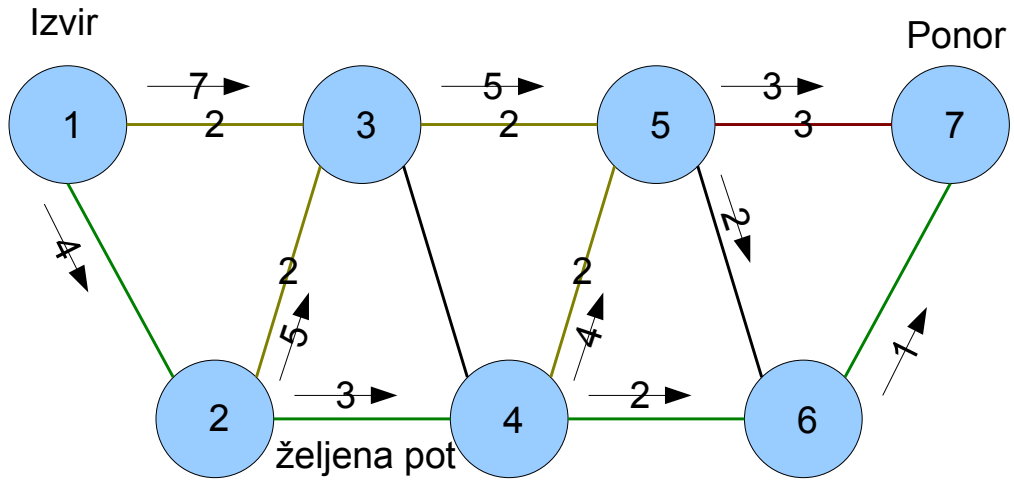


Slika 2.2: Optimalna pot in željena pot.

vidi končno destinacijo, dosegljivo preko usmerjevalnika 7, s ceno 1, preko usmerjevalnika 6 pa s ceno 2. Cena skozi 6 mora biti bolj ugodna, zato moramo utežiti povezavo $5-7$ s ceno 3. Na sliki 2.3 vidimo uteži na povezavah (številke na povezavah) ter cene poti do usmerjevalnika 7 (številke na puščicah).

2.1.2 Druga naloga

Za naslednjo nalogo izberimo željeno pot $1-2-3-4-5-6-7$. Naša želja v tem primeru je, da gre promet v normalnem stanju (brez izpadov povezav ali usmerjevalnikov) po željeni poti, potek prometa pri izpadih pa nas ne zanima. Za zagotovitev takega poteka prometa moramo utežiti povezave $1-3$, $3-5$, $5-7$, $2-4$ in $4-6$. Tokrat moramo vsem dodeliti metriko 3. Zakaj je to treba, si lahko pogledamo na delu omrežja. Če gledamo trikotnik povezav med usmerjevalniki $2, 3$ in 4, potem vidimo, da je željena pot $2-3-4$, pot $2-4$ pa je brez uteži optimalna in sestoji iz ene same povezave. Če bi to povezavo utežili z metriko 2, bi bili obe poti enakovredni, česar pa nočemo. Zato jo moramo utežiti z metriko 3. Analogno velja za ostale trikotnike $(1, 2, 3)$, $(3, 4, 5)$, $(4, 5, 6)$ in $(5, 6, 7)$ v grafu. Slika 2.4 prikazuje uteži ter cene v grafu povezav, ki predstavljajo rešitev druge naloge.



Slika 2.3: Uteži za pot 1-2-4-6-7.

2.1.3 Tretja naloga

Za zadnjo nalogo izberimo naslednje parametre:

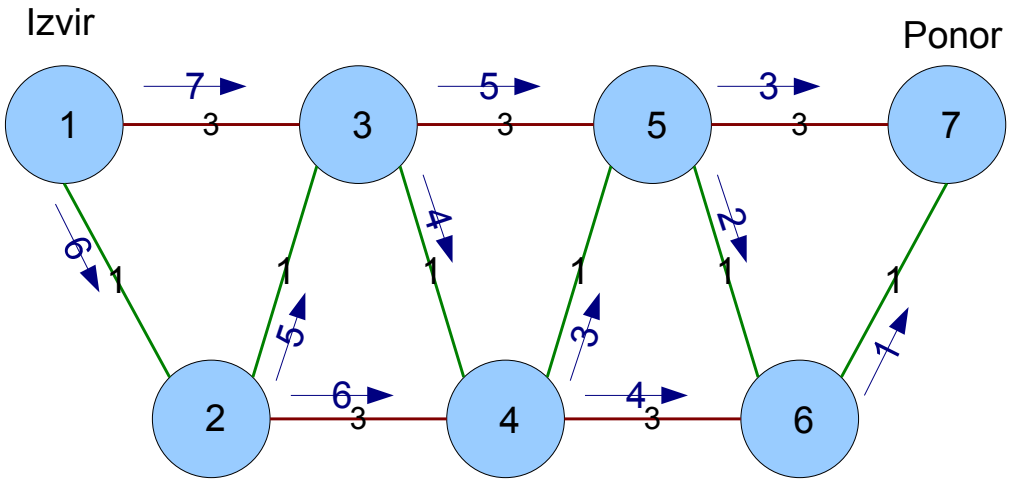
- želimo, da promet teče po isti cikcakasti povezavi kot v prejšnji nalogi,
- če pride do odpovedi povezave, naj se promet za vsako ceno izogiba zgornji poti $(1-3-5-7)$; to naj izbere le, če ni drugih možnosti.

Za osnovo lahko vzamemo že utežene povezave prejšnje naloge. Dodatne uteži bodo služile izogibanju zgornji poti. Potrebne spremembe so naslednje:

- povezava $5-7$ mora biti utežena z metriko 6, da usmerjevalnik 5 v vsakem primeru izbere pot $5-4-6-7$ (s ceno 5, ker ima povezava $4-6$ metriko 3) namesto poti $5-7$,
- povezava $3-5$ mora biti utežena z metriko 4, da usmerjevalnik 3 izbere pot do 7 preko usmerjevalnika 4 in ne preko usmerjevalnika 5.

2.1.4 Zaključki

- pri zadnji nalogi imamo na treh skokih metriko že na $3+4+6=13$; to predstavlja povečanje za faktor $4,3$; pri štirih hopih bi že presegli



Slika 2.4: Uteži za cikcakasto pot.

največjo možno metriko pri RIP-u, ki je 15; alternativni protokoli, ki nimajo te omejitve, so:

OSPF in ISIS, ki kot pripadnika družine "link-state" ne omogočata nastavljanja metrike za posamezne ciljne destinacije (podomrežja), temveč le za povezavo in sta tako bistveno manj fleksibilna pri upravljanju prometa,

EIGRP, ki je podprt le na usmerjevalnikih proizvajalca Cisco,

BGP, ki ni namenjen usmerjanju znotraj avtonomnih sistemov in bi povečal kompleksnost nastavljanja usmerjevalnikov;

- kompleksnost nastavljanja usmerjevalnikov raste s kompleksnostjo omrežja; spremembe metrike na povezavah je treba izračunati, ti izračuni pa niso intuitivni in s tem povečajo možnost za napako;

- spremembe vplivajo le na usmerjanje prometa proti nekem cilju; če bi želeli različne izbrane poti med različnimi viri prometa do istega ponora, tega ni možno realizirati.

2.2 Uporaba tunelskih povezav namesto manipulacije metrike

Ker je predhodno omenjena manipulacija metrik kompleksna in težavna v praksi, bi lahko uporabili tunelske povezave kot premostitveni element. Tunelska povezava je navidezna povezava med dvema točkama, ki sta predstavljeni z naslovi IP. S tem lahko poljubna dva usmerjevalnika, ki sta dosegljiva drug drugemu preko nekega omrežja, prikažemo kot direktno povezana. Ves promet, ki je namenjen skozi tunelsko povezavo, se ovije v nov protokol, ponavadi GRE. Doda se nova glava protokola IP, kjer imamo nov izvorni in ponorni naslov. Ker vmesnik, ki predstavlja tunelsko povezavo, za usmerjevalni protokol ni nič drugačen od fizičnih, se za tunelsko povezavo izračuna metriko tako kot za vsako drugo povezavo. Za lažjo manipulacijo naredimo za vsako stran tunela povratni vmesnik (angl. *loopback interface*), ki se obnaša kot kateri koli drug vmesnik na usmerjevalniku, le da ni vezan na fizično povezavo in je tako vedno aktiven (običajni vmesniki postanejo neaktivni, če izgubijo fizično povezljivost).

Ovit ali enkapsuliran promet imenujmo tunelski promet, promet, ki je namenjen skozi tunelske povezave, pa podatkovni promet.

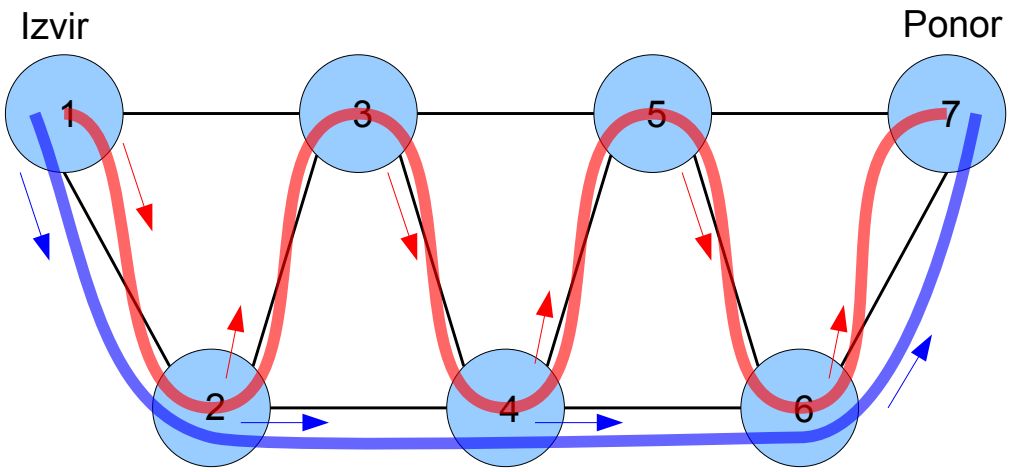
Če vzamemo za primer topologijo iz prejšnje sekcije in želimo implementirati identična pravila, potem se zadnjo nalogo realizira na naslednji način:

Ustvarimo dve tunelski povezavi, ki nam predstavljata prvo $(1-2-3-4-5-6-7)$ in drugo $(1-2-4-6-7)$ željeno pot. Obe tunelski povezavi vidimo na sliki 2.5. Sedaj moramo fiksirati tunela na to pot. To naredimo tako, da na vseh usmerjevalnikih nastavimo statične poti za povratni vmesnik, kjer se zaključuje tunel. Tako imamo na usmerjevalnikih od 1 do 6 statične poti, ki kažejo proti naslednjemu usmerjevalniku v vrsti za ciljni naslov prvega tunela, ter na usmerjevalnikih 1, 2, 4 in 6 statične poti za ciljni naslov drugega tunela.

S tem smo pridobili dve novi navidezni povezavi med usmerjevalnikoma 1 in 7. Ti dve povezavi se, kot že omenjeno, lahko uporabljata popolnoma enakovredno vsem ostalim povezavam. Če aktiviramo usmerjevalni protokol na usmerjevalnikih 1 in 7, se bosta videla, vzpostavila sosednost in usmerila promet skozi tunelske povezave. S tem si zelo poenostavimo manipulacijo metrik na usmerjevalnikih.

Implementacija takih tunelov pa prinese dva problema:

1. Prva stvar, ki se lahko izkaže za problematično, je fiksnost tunelov, kar pomeni naslednje: če izgubimo katero od fizičnih povezav na poti tunela, zgubimo celotno tunelsko povezavo. Tunel je fiksiran na točno specifične



Slika 2.5: Tunelski povezavi.

fizične povezave s statičnimi potmi. Ta problem lahko rešimo na dva načina, noben pa žal ni eleganten.

(a) Prva rešitev je kreiranje alternativnega tunela, kar smo tudi naredili v rešitvi zadnje naloge - tunel $1-2-4-6-7$ je alternativni tunel. Če bi odpovedal tudi alternativni tunel, je za ohranjanje povezljivosti priporočljivo, da vsi usmerjevalniki na poti omogočajo usmerjanje podatkovnega prometa in ne samo tunelskega. Na ta način si ohranimo vsaj možnost usmerjanja prometa, če bi slučajno vse tunelske povezave zaradi odpovedi fizičnih povezav postale neuporabne.

(b) Druga rešitev je uporaba dinamičnega usmerjanja za naslov, kamor je pripet tunel. Vendar s tem pridemo nazaj na prvotni problem, ki smo ga hoteli rešiti v prejšnji sekciji, le da zdaj usmerjamo tunelski in ne podatkovnega prometa.

2. Druga težava, ki jo ustvarimo, je vezana na ovijanje. Za začetek imamo vpliv na performanse usmerjevalnika. Dodatno ovijanje zahteva dodatno procesiranje. Dodatna glava zmanjša število oktetov, ki jih lahko namenimo podatkovnemu prometu, in zmanjša prepustnost povezav.

2.3 Deljenje bremena

Ena izmed možnosti, ki nam omogočajo izkoriščanje več kot ene povezave, je deljenje bremena (angl. *load balancing* ali *load sharing*). Deljenje bremena v kontekstu mrež pomeni, da se za neko ciljno podomrežje pošilja promet na več kot le eno povezavo. Obremenitev si deli več povezav in s tem razbremenijo tisto eno, optimalno.

Ker še vedno uporabljamo klasično usmerjanje, delitev prometa na več povezav dosežemo s tem, da usmerjevalnemu protokolu prikažemo več povezav oziroma poti, enakovrednih med seboj. Skupna metrika več poti do ciljnega podomrežja mora biti enaka. S tem se usmerjevalnik odloči za obe poti oziroma povezavi hkrati. Poleg tega mora biti ta metrika tudi najnižja in s tem najbolj ugodna.

Ko usmerjevalnik izbere dve izhodni povezavi kot enakovredni, začne pošiljati promet skozi ti povezavi. Pri tem se lahko na več različnih načinov odloči, kdaj bo poslal nek paket skozi eno in kdaj skozi drugi povezavo. Najbolj pogosto uporabljeni načini so naslednji [1]:

- "per packet"; pri tem načinu usmerjevalnik uporablja "round-robin" algoritem za izbiro povezave, glede na vsak paket, ki ga mora poslati na ciljno podomrežje,

- "per source/destination pair"; pri tem načinu se paketi iz nekega izvornega naslova na nek ponorni naslov vedno obravnavajo enako (layer-3 information) - to pomeni, da se pošljejo vedno skozi isto izhodno povezavo,

- "per flow"; pri tem načinu se obravnavajo enako vsi paketi neke seje (layer-4 information); tu mora usmerjevalnik spremljati več podatkov kot pri prejšnjem načinu in ima lahko zaradi tega negativen vpliv na performanso.

Zagotovitev enake metrike ima enake zahteve kot opisane v preusmerjanju z metriko.

Prednost deljenja prometa je, da nam na kratih razdaljah (poteh z malo povezav) omogoča preprosto utilizacijo več povezav in s tem razbremeni potencialna ozka grla v omrežju. Slabosti je precej več, in sicer:

- metrika različnih poti mora biti enaka,

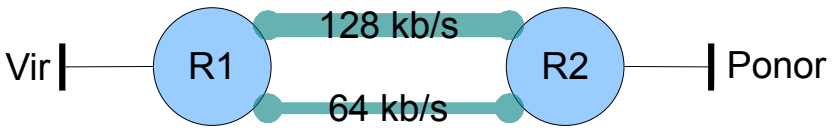
- če imajo povezave različne kapacitete, bodo nekatere prej preobremenjene kot druge, ker se bo obremenitev prerazporedila enakomerno čez vse poti,
- deljenje prometa "per packet" ima lahko zelo slab učinek na peformanso (glej razdelek 2.3.1),
- če delimo glede na informacijo tretjega oz. četrtega nivoja in se lahko zgodi, da imamo le eno požrešno sejo oziroma en par govorcev, bomo še vedno obremenili le eno pot (samo zaporedne povezave na poti, nikoli vzporednih),
- ker uporabljamo klasično usmerjanje, prometa ne moremo poljubno usmerjati glede na izvor.

2.3.1 Deljenje prometa po paketih

Prvi problem pri razporejanju paketov na več kot en vmesnik, pri čemer vsak paket obravnavamo individualno, je v prihajanju paketov na cilj v napačnem zaporedju (angl. *out of order delivery*). To se zgodi zaradi različnih zakasnitev na povezavah. Ciljna naprava mora pakete prerazporediti v pravilni vrstni red (angl. *TCP reorder*). Problem preurejanja paketov se nanaša le na protokol TCP, ki aplikacijam zagotavlja dostavo v pravem vrstnem redu. Protokol UDP tega mehanizma ne pozna, kar pa ne pomeni, da nam napačno zaporedje paketov ne bo povzročalo težav. Primer je kakršen koli govorni promet, kjer se napačno dostavljene pakete preprosto odmetava, ali pa protokol TFTP, ki se uporablja za prenašanje datotek, povečini na preprostejših platformah, kjer bi bila implementacija protola TCP neprimerna. Drug primer bi bila aplikacija, ki sama izvaja prerazporejanje paketov.

Drugi problem pri protokolu TCP je potrjevanje paketov. V primeru dveh povezav različnih kapacitet teoretično ne moremo doseči večje prepustnosti, kot je dvakratnik manjše kapacitete. Razlog za to je, da tudi če lahko prva povezava prenese veliko več paketov v danem časovnem intervalu kot druga povezava, bo usmerjevalnik vsak drugi paket poslal skozi drugo povezavo. Če vzamemo nek interval paketov, ciljna naprava ne bo potrdila zadnjega paketa v intervalu, dokler ne bo prejela tudi vseh predhodnih paketov. Zaradi tega nam ne koristi, če ima ena od povezav večjo prepustnost, ker vedno čakamo na pakete, potujoče skozi počasnejšo povezavo.

V praksi je rezultat slabši od teoretičnega. Kot dokaz vzemimo naslednji primer.



Slika 2.6: Testno omrežje za testiranje deljenja prometa.

Primer: *Testno okolje sestoji iz dveh usmerjevalnikov, na prvem je vir prometa, na drugem je ponor prometa. Povezavi med usmerjevalnikoma sta dve, prva ima prepustnost 128 kbit/s, druga 64 kbit/s. Vir in ponor sta računalnika. Prvi bo pošiljal pakete, drugi jih bo sprejemal. Shemo lahko vidimo na sliki 2.6. Rezultat izkoriščanja izključno hitrejše povezave vidimo v tabeli 2.1.*

Tabela 2.1: Prenos skozi hitro povezavo.

tek	število oktetov	čas	pretok
prvi tek:	409 600	26,7 s	123 kb/s
drugi tek:	409 600	26,7 s	123 kb/s

Ko je bil promet preusmerjen na počasnejšo povezavo, se je prenos po pričakovanju zmanjšal. Rezultat je predstavljen v tabeli 2.2.

Ob deljenju prometa so rezultati precej slabši od teoretičnega maksimuma, ki je seštevek obeh poti. V praksi se izkaže, da skupen prenos skozi obe povezavi ne dosega niti pretoka ene same hitre povezave. Rezultati so prikazani v tabeli 2.3.

Rezultat je očitno slabši od izkoriščanja le ene (boljše) povezave.

Tabela 2.2: Prenos skozi počasnejšo povezavo.

tek	število oktetov	čas	pretok
prvi tek:	307 200	40,1 s	61,4 kb/s
drugi tek:	307 200	40,1 s	64,3 kb/s

Tabela 2.3: Prenos skozi obe povezavi hkrati.

tek	število oktetov	čas	pretok
prvi tek:	307 200	26,1 s	94,1 kb/s
drugi tek:	307 200	25,8 s	95,2 kb/s
tretji tek:	307 200	26,4 s	93,1 kb/s

2.3.2 Deljenje bremena pri različnih metrikah s protokolom EIGRP

Proizvajalec Cisco ima v svojem zaprtem protokolu EIGRP možnost deljenja prometa preko več povezav z različnimi metrikami. Uporablja količnik, ki se mu reče varianca in ga nastavi uporabnik. Ko usmerjevalnik izračunava možne poti, ki jih bo uporabil za pošiljanje prometa, vzame metriko poti, ki jo trenutno ocenjuje in metriko najboljše poti. Če je metrika poti, ki jo gledamo, manjša od najboljše metrike, pomnožene z varianco, se jo uporabi za pošiljanje prometa. Promet se deli glede na razmerje metrik. Vse vrednosti (varianca in količniki) se zaokrožujejo na celoštevilske vrednosti [2].

Primer:

- *optimalna pot ima metriko 50 (M_{min}),*
- *vse ostale poti imajo metriko, večjo od 50,*
- *varianca (v) je nastavljena na 3 (zato je mejna vrednost metrike $M_m = 3*50 = 150$),*
- *imamo druge poti z metrikami: 70, 110, 160 in 200,*

Tabela 2.4: Deljenje bremena z varianco s protokolom EIGRP.

pot i	metrika M_i	količnik $K_i = \frac{M_m}{M_i}$	delež prometa $D_i = \frac{K_i}{\sum_j K_j}$
1	50	3	1/2
2	70	2	1/3
3	110	1	1/6
4	160	0	0
5	200	0	0

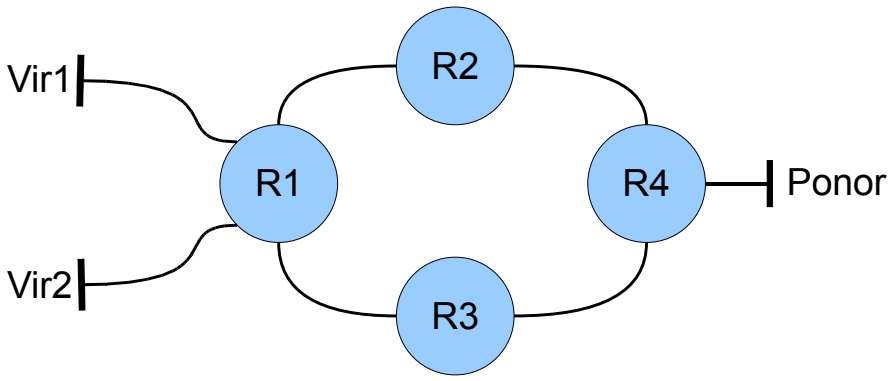
V tabeli 2.4 vidimo, da bo usmerjevalnik poslal 1/2 prometa na optimalno pot, 1/3 na pot z metriko 70 in 1/6 na pot z metriko 110. Varianca pri EIGRP nam izjemno poenostavi deljenje prometa na več povezav. Vendar je to le en rešen problem. Še vedno so tu slabosti deljenja prometa po paketih oziroma zahteve po veliko sejah, če se odločimo za deljenje glede na informacije z višjih nivojev (tretjega in četrtega nivoja). Poleg tega smo vezani na enega izdelovalca opreme in na en protokol.

2.4 Usmerjanje prometa glede na izvorni naslov z usmerjanjem s pravili

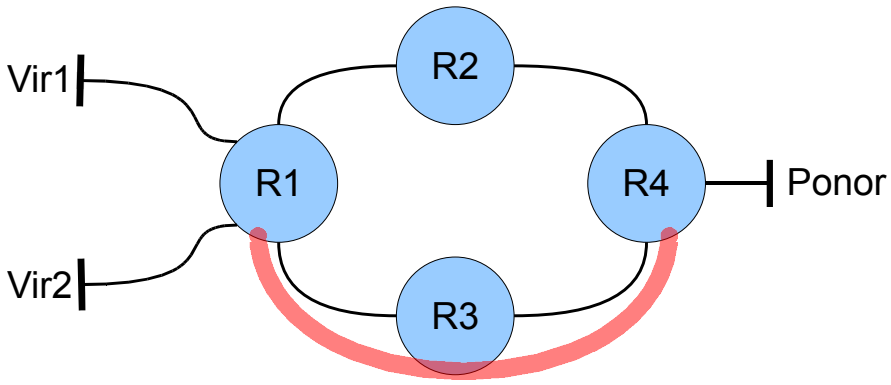
Strategija "policy routing" nam omogoča, da usmerjamo promet glede na definirano politiko, za razliko od običajnega usmerjanja, kjer se promet vedno usmerja glede na ciljni naslov [3]. Dva najbolj pogosta načina uporabe tega mehanizma sta usmerjanje glede na izvorni naslov, kjer ciljni naslov ne igra vloge pri odločanju, kam se bo promet usmeril, ter usmerjanje prometa mimo klasičnih usmerjevalnih protokolov. Največji problem pri uporabi usmerjanja s pravili je fiksiranje odločitve. Pri dinamičnih usmerjevalnih protokolih se lahko usmerjanje prilagaja stanju v omrežju. To je tudi osnovni namen dinamičnega usmerjanja. Usmerjanje s pravili pa je enako statičnim potem. Če pride do odpovedi opreme nekje na poti, bomo izgubili ves promet, ki smo ga usmerili na to pot.

Če je naš namen usmeriti del prometa na drugo pot, potem lahko uporabimo trik s tuneli. Slika 2.7 prikazuje topologijo omrežja, ki ga bomo uporabili v primeru. Omrežje ima dva vira prometa, štiri usmerjevalnike in en ponor.

Ves promet gre po isti poti, ker se usmerja le glede na ciljni naslov. Če bi želeli razdeliti promet na dve poti, je prvi korak dodatna tunelska povezava



Slika 2.7: Topologija omrežja z dvema viroma prometa.



Slika 2.8: Topologija omrežja z dvema viroma prometa z dodatno tunelsko povezavo.

med R1 in R4, ki jo lahko vidimo na sliki 2.8. S tunelsko povezavo dobimo dva nova vmesnika na R1 in R4. Če zdaj usmerimo promet z vira 1 proti ponoru s pravilom, da mora ves promet, ki izvira iz naslovnega prostora na

viru 2, iti skozi povezavo "tunel", smo s tem obdržali možnost preusmeritve. Tuneliran promet se obravnava kot vsak drug promet in je zato usmerjen glede na ciljni naslov. Ker pa je ciljni naslov za tunel fiksiran, različen od ostalih naslovov, ter dosegljiv tako preko R2 kot preko R3, smo problem usmerjanja glede na izvorni naslov prevedli na problem izbiranja alternativne poti za nek ciljni naslov.

2.5 Izvorno usmerjanje prometa

IP podpira možnost definiranja vmesnih usmerjevalnikov na poti vsakega paketa [5]. Teoretično bi bilo možno implementirati nastavitev tunelskega vmesnika, da usmerjevalnik vse pakete, ki so enkapsulirani v nek tunel, opremi z informacijo, preko katerih vmesnih usmerjevalnikov morajo potovati. Če vsi vmesni usmerjevalniki podpirajo ta način in če ni nobenih varnostnih mehanizmov, ki bi omejevali pakete s to informacijo, je možno vsak tunel preusmeriti čez poljubne usmerjevalnike. V praksi na usmerjevalnikih tega ni možno nastaviti.

Poglavje 3

Rešitev z uporabo prometnega načrtovanja

V prejšnjem poglavju smo predelali različne načine kontroliranja prometa s klasičnimi metodami. V tem poglavju si bomo ogledali kontrolo s pomočjo prometnega načrtovanja. Prometno načrtovanje v okolju MPLS je omogočeno s tunelskimi povezavami. Bistvo MPLS-a je odločanje, v katero smer pošljemo promet na osnovi oznak, ki jim v žargonu rečemo labele (angl. *Labels*). Od tu pride tudi ime MPLS (angl. *Multi Protocol Label Switching*).

3.1 Osnove MPLS-a

MPLS pošilja promet na sosednje naprave glede na labele, ki se izmenjajo med napravami. Tako kot klasično usmerjanje tudi MPLS usmerja promet glede na ciljne naslove. Klasični notranji usmerjevalni protokoli so še vedno nujno potrebni pri uporabi MPLS-a, tako da MPLS ne predstavlja alternative za usmerjanje, temveč le za hitro izbiranje izhodne poti za posamezne pakete. Če se postavimo na enega izmed usmerjevalnikov v oblaku MPLS-a, ta usmerjevalnik vsakemu ciljnemu podomrežju D, ki je prisotno v usmerjevalni tabeli, dodeli labelo L (celoštevilsko vrednost). Svojim sosedom nato posreduje to labelo (s protokolom TDP ali LDP), kar sosedje razumejo kot "Pozdravljen, če želiš meni poslati promet, katerega končna destinacija je D, označi ta promet z labelo L.". Naš izbrani usmerjevalnik dobi podobna sporočila, ki povezujejo ciljna podomrežja z labelami, tudi od svojih sosedov. Zato mora vsakemu paketu zamenjati labelo, s katero je bil označen, ko je prispel. S tem spreminjanjem label (angl. *Label Switching*) se definira pot prometa skozi omrežje. Tako pot prometa imenujemo LSP (angl. *Label Switched Path*). Obstaja za

vsak promet, ki vstopi in zapusti oblak MPLS. Prometno načrtovanje v okviru MPLS-a se ukvarja predvsem z izbiro LSP-ja, ki morda ni optimalen s stališča klasičnega usmerjanja, je pa zaželjen z naše strani, zaradi boljšega izkoristka omrežnih resursov. Kot že omenjeno, se izbira drugačnega LSP-ja izvrši z izgradnjo tunela.

3.2 Tuneli

Za razliko od klasičnih tunelov, ki jih sicer uporabljamo v svetu omrežij in so obojesmerni (vsak tunel ima dva konca, promet lahko pošljemo v tunel na kateri koli strani in ven bo prišel na drugi strani), so tuneli v svetu MPLS-a enosmerni. Imajo glavo in rep, promet pa poteka le v eno smer (nemogoče je poslati promet od repa proti glavi). Namen tunelov TE je definicija neke nove poti skozi oblak MPLS. Ta pot bo drugačna od privzete poti ostalega prometa. Kateri promet bo ubral to drugačno pot, je odločitev usmerjevalnika na začetku tunela (pri glavi). Najpomembnejša stvar, ki tunele TE naredi zelo uporabne, je eksplicitna definicija željene poti. Imamo možnost definiranja celotne poti, dela poti - naštejemo lahko usmerjevalnike, preko katerih promet mora iti, ali pa dele omrežja, ki se mu mora promet izogibati (zopet lahko naštejemo usmerjevalnike, v tem primeru tiste, preko katerih promet nikakor ne sme iti). Ta funkcionalnost nam omogoča fleksibilnost pri izbiri poti, hkrati pa je enostavnost implementacije bistveno večja kot pri enakem dosežku s postopkom, opisanim v poglavju 2.1. Poleg tega, da lahko fiksiramo pot, lahko tudi naštejemo več poti z različnimi prioritetami. Za razliko od pristopa, ki smo ga ubrali v poglavju 2.1, kjer se izbira pot na osnovi metrike poti, tu pot definiramo ne s povezavami, temveč z napravami. To naredi izbiranje poti nekoliko drugačno. Če si zopet izberemo topologijo iz poglavja 2.1 in rešujemo problem tretje naloge, potem je najboljša možnost, da naštejemo željene poti v prioritetnem vrstnem redu (najbolj zaželjena pot najvišje).

- $1-2-3-4-5-6-7$,
- $1-2-3-4-6-7$ ali $1-2-4-5-6-7$,
- $1-2-4-6-7$ in
- poljubna pot.

Alternativen način določanja željenih poti je s specificiranjem prepovedanih usmerjevalnikov oziroma povezav. Ker imamo v našem primeru zaželjene in

nezaželjene poti preko istih usmerjevalnikov, je treba izločiti posamezne povezave in ne usmerjevalnikov. Ko nastavimo neko povezavo za nezaželjeno, promet nikoli ne bo šel preko nje. Zato je treba določiti vse možne kombinacije legitimnih poti, kar lahko pomeni precejšnje število poti, tako kot je to v našem primeru.

3.3 Metrika

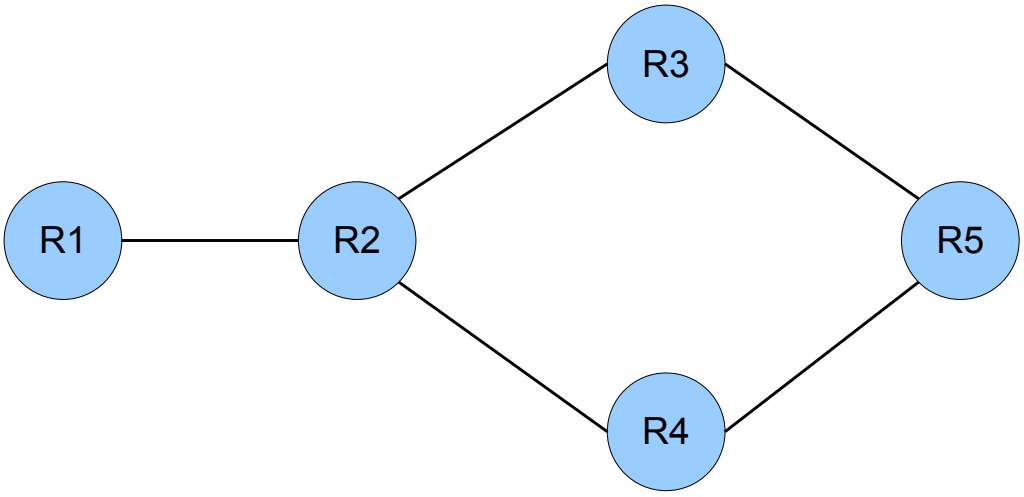
Povsem druga možnost, ki jo imamo na voljo, je nastavitev prometno inženirske metrike. Z njo ne določamo poti, ampak le vplivamo na izbiro povezav pri dinamičnih poteh ali pri nedefiniranih delih fiksiranih poti. Pomembno pa je razumeti dve stvari, ki se tičeta te metrike. Ta metrika se uporablja pri vseh tunelih - je torej globalna nastavitev v svetu prometnega inženiringa - ne moremo je nastaviti le za en tunel. Druga stvar je, da se ta metrika uporablja le za dinamične dele poti. Ko enkrat pot eksplicitno določimo, bo tunel potekal po tisti poti ne glede na metriko [4].

3.4 Optimizacija poti

Ko nastavljamo usmerjevalnike ter tunele, je eden izmed korakov, da določimo kapacitete povezav. Določiti moramo tako kapacitete fizičnih povezav kot kapacitete tunelskih povezav. Fizične povezave moramo nastaviti zato, da usmerjevalniki vedo, kolikšne kapacitete imajo na razpolago. Kapacitete tunelskih povezav nam po drugi strani povejo, koliko razpoložljivih kapacitet želimo rezervirati za naše poti.

Prometni inženiring žal ne omogoča optimizacije vseh tunelskih povezav na dani topologiji. Če imamo topologijo, kot je prikazana na sliki 3.4, potem sta dve možni poti od R1 do R5. Naj bo kapaciteta poti preko R3 50 kb/s, kapaciteta preko R4 pa 20 kb/s. Če želimo med R1 in R5 vzpostaviti dve tunelski povezavi, prvo s kapaciteto 15 kb/s, drugo pa s kapaciteto 45 kb/s, potem je to očitno možno, vendar če bi povezavi aktivirali v tem vrstnem redu, brez drugih omejitev, bi se prva tunelska povezava vzpostavila preko R3, za drugo pa ne bi bilo več prostih kapacitet.

Da se izognemo takim nevšečnostim, lahko tunelskim povezavam priredimo prioritetne željene poti (v našem primeru gre prva tunelska povezava preko R4, druga pa preko R3). V primeru odpovedi naše tunelske povezave nimajo alternativnih poti zaradi premajhnih kapacitet fizičnih povezav. Če bi te vendarle obstajale, bi morali tunelskim povezavam uporabo teh poti dovoliti (kot smo to



Slika 3.1: Omrežje z dvema potema do cilja.

nakazali v poglavju 3.2). Ob ponovni vzpostavitvi primarne poti se tunelska povezava ne bo samodejno prestavila nanjo, če ne vklopimo funkcionalnosti reoptimizacije. Reoptimizacija v terminologiji prometnega inženiringa pomeni le, da se ob specifičnih trenutkih (dogodek, časovni interval) sproži proces, ki preveri, če obstaja boljša pot za tunelske povezave, kot so trenutno izbrane [8].

Poglavje 4

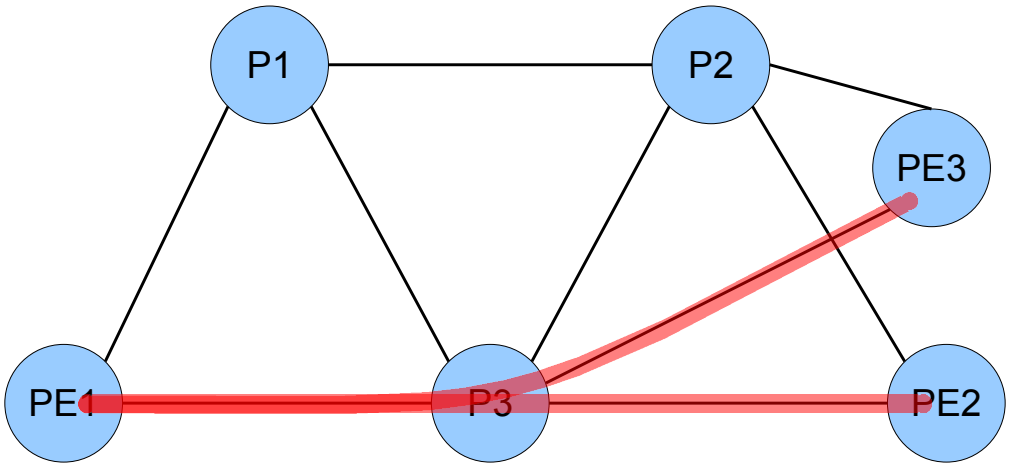
Primerjava klasične tunelske povezave in MPLS TE pristopa

Kontrola prometa z uporabo tehnologije MPLS TE oziroma kakršna koli kontrola prometa je primarno zaželjena v omrežjih ponudnikov komunikacijskih storitev. Ta omrežja sestojijo iz usmerjevalnikov, ki jih delimo na dve vrsti - usmerjevalniki ponudnika (angl. *Provider* - krajše P) ter robni usmerjevalniki ponudnika (angl. *Provider Edge* - krajše PE). Usmerjevalniki P sestavljajo hrbtenico omrežja, katere osnovna naloga je zagotavljanje visoko redundančnih povezav med robnimi usmerjevalniki. Ponudniki svojih strank navadno ne priključujejo neposredno na hrbtenično omrežje. Stranke priključujejo na robne usmerjevalnike. Naloga usmerjevalnikov PE je zagotavljanje povezljivosti strank na hrbtenično omrežje. Navadno imajo več različnih tipov vmesnikov in podpirajo različne storitve, ki so potrebne za priklop strankinih usmerjevalnikov [7].

V tem poglavju bomo primerjali dva pristopa k optimizaciji prometnih tokov v omrežju. Pristop z uporabo tunelov MPLS TE bomo primerjali z najboljšo izbiro alternativnih metod, ki smo jih primerjali v prejšnjih poglavjih. Za alternativno metodo izberimo klasične tunelske povezave s fiksiranimi tuneli, saj bo tudi tunel MPLS TE fiksiran. Hkrati ta metoda omogoča najbolj fleksibilno izbiranje poti, z najmanj dodatne konfiguracije. Namen primerjave je ugotoviti, katera metoda je bolj fleksibilna in kako kompleksnost konfiguracije raste z večanjem omrežja.

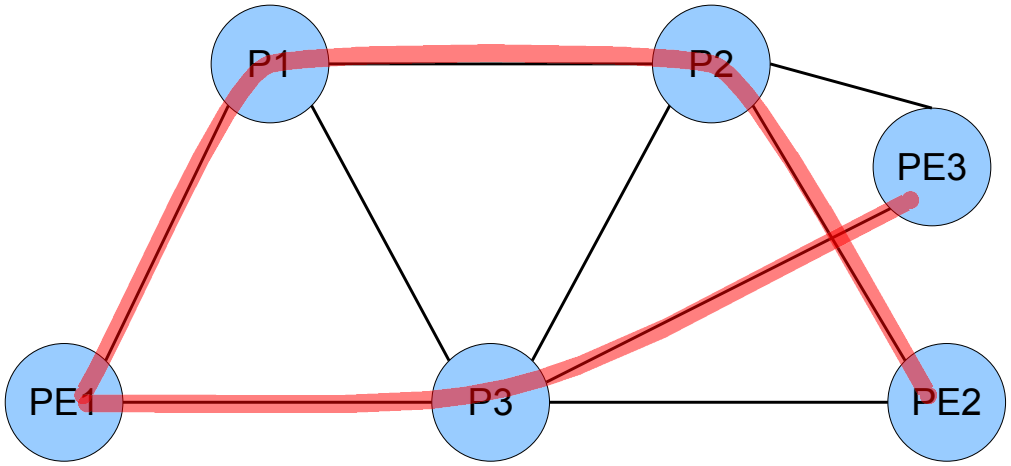
4.1 Opis in parametri testnega omrežja

Za potrebe primerjave si vzemimo enostavno omrežje, ki sestoji iz treh hrbteničnih usmerjevalnikov ter treh robnih usmerjevalnikov. Topologijo vidimo na sliki 4.1. Na levi strani imamo robni usmerjevalnik PE1, na katerega priključujemo stranke. Enako imamo na desni strani dva robna usmerjevalnika PE2 in PE3, kamor priključujemo stranke. V našem hipotetičnem okolju je analiza pokazala, da je med robnim usmerjevalnikom PE1 ter robnima usmerjevalnikoma PE2 in PE3 veliko prometa, kar obremenjuje povezavo PE1−P3. Vsi usmerjevalniki PE so zaradi redundance na hrbtenično omrežje priključeni z dvema povezavama. Naš načrt je preusmeritev dela prometa s povezave PE1−P3 na povezavo PE1−P1 in od tam dalje.



Slika 4.1: Topologija in privzeta pot prometa.

Usmerjevalnike testnega omrežja moramo nasloviti z naslovi IP. Vsak usmerjevalnik ima svoj naslov za vmesnik “loopback”, nasloviti pa moramo tudi vse povezovalne segmente. V naslednjih tabelah so našteti naslovi vseh usmerjevalnikov in njihovih povezovalnih segmentov. Tabela 4.1 vsebuje naslove hrbteničnih usmerjevalnikov (tip P). Tabela 4.2 vsebuje naslove robnih usmerjevalnikov (tip PE). Tabela 4.3 vsebuje naslove povezovalnih segmentov med usmerjevalniki.



Slika 4.2: Željena pot prometa - drugačna za promet med PE1 in PE2.

Tabela 4.1: Naslovi hrbteničnih usmerjevalnikov.

Usmerjevalnik	Naslov
P1	10.0.0.1
P2	10.0.0.2
P3	10.0.0.3

Tabela 4.2: Naslovi robnih usmerjevalnikov.

Usmerjevalnik	Naslov	Vmesnik za tunelske povezave
PE1	10.0.1.1	10.0.2.1
PE2	10.0.1.2	10.0.2.2
PE3	10.0.1.3	10.0.2.3

Tabela 4.3: Naslovi povezovalnih segmentov.

Povezava	Podomrežje povezave	Naslov prvega usmerjevalnika	Naslov drugega usmerjevalnika
P1 – P2	10.0.12.0/24	10.0.12.1	10.0.12.2
P1 – P3	10.0.13.0/24	10.0.13.1	10.0.13.3
P2 – P3	10.0.23.0/24	10.0.23.2	10.0.23.3
PE1 – P1	10.0.111.0/24	10.0.111.11	10.0.111.1
PE1 – P3	10.0.113.0/24	10.0.113.13	10.0.113.1
PE2 – P2	10.0.122.0/24	10.0.112.12	10.0.112.2
PE2 – P3	10.0.123.0/24	10.0.123.12	10.0.123.3
PE3 – P2	10.0.132.0/24	10.0.132.13	10.0.132.2
PE3 – P3	10.0.133.0/24	10.0.133.13	10.0.133.3
Tunnel1	10.0.201.0/24	10.0.201.1	10.0.201.2

4.2 Konfiguracije usmerjevalnikov s klasično tunelsko povezavo

Dodatna konfiguracija na vseh usmerjevalnikih je podana s kodo:

```
ip route 10.0.2.2 255.255.255.255 A.B.C.D
ip route 10.0.2.1 255.255.255.255 E.F.G.H
```

Naloga teh dveh vrstic je usmeriti tunelski promet na pravo pot. Pri klasičnih tunelih je takšno usmerjanje edini možni način usmerjanja prometa. Skupaj je treba dodati dve vrstici za vsako tunelsko povezavo, ki prečka ta usmerjevalnik.

Na robnih usmerjevalnikih PE1 in PE2 je treba definirati tunelsko povezavo:

```
interface loopback1
 ip address 10.0.2.1 255.255.255.255
interface tunnel1
 ip address 10.0.201.1 255.255.255.252
 tunnel destination 10.0.2.2
 tunnel source loopback1
 tunnel mode ipip
```

```
interface loopback 1
 ip address 10.0.2.2 255.255.255.255
interface tunnel 100
 ip address 10.0.201.2 255.255.255.252
 tunnel destination 10.0.2.1
 tunnel source loopback1
 tunnel mode ipip
```

Na obeh usmerjevalnikih, med katerima poteka tunelska povezava, naredimo dva vmesnika. Vmesnik tipa "loopback" nam služi za pripetje tunelske povezave. Promet proti temu vmesniku usmerjamo s statičnimi usmeritvami, kot so prikazane v prejšnjem primeru konfiguracije. Vmesnik tunnel1 je vmesnik virtualne tunelske povezave. Začetek in konec (angl. *source* in *destination*) povesta, kam je tunelska povezava pripeta. S tipom enkapsulacije (angl. *mode ipip*) določimo najbolj enostaven način ovijanja prometa.

Potrebno je tudi na nek način usmeriti promet v novo tunelsko povezavo. To lahko storimo s statično potjo ali pa z dinamičnim usmerjanjem, kjer vključimo tunelsko povezavo v usmerjevalni protokol. V obeh primerih je potrebna le ena dodatna vrstica konfiguracije. V našem primeru bomo uporabili dinamično usmerjanje, da bosta funkcionalno obe rešitvi čim bolj podobni.

4.3 Konfiguracije usmerjevalnikov z MPLS TE pristopom

Usmerjevalniki P in PE morajo za potrebe MPLS TE imeti vključene naslednje funkcionalnosti:

- podpora za MPLS TE na vmesnikih (1 vrstica in 1 vrstica na vmesnik),
- podpora za MPLS TE v okviru usmerjevalnega protokola (2 vrstici),
- podpora za RSVP (1 vrstica na vmesnik).

Na usmerjevalnikih PE moramo definirati tunelske povezave, ki jih želimo uvesti v omrežje. Usmerjevalniki P ne potrebujejo nobene informacije o tunelskih povezavah. Prednost tega je, da rast hrbteničnega omrežja ne vpliva na kompleksnost konfiguracije.

Dodatna konfiguracija na vseh usmerjevalnikih za potrebe MPLS TE je naslednja:

```
mpls traffic-eng tunnels
interface gigabitethernet X/Y
 mpls traffic-eng tunnels
 ip rsvp bandwidth 100000
router ospf 1
 mpls traffic-eng router-id loopback0
 mpls traffic-eng area 0
```

Skupaj je to $3 + 2 * N$ dodatnih vrstic, pri čemer je N število vmesnikov v hrbteničnem omrežju.

Na robnem usmerjevalniku PE1 moramo definirati novo tunelsko povezavo in ji določiti pot. Konfiguracija je naslednja:

```
interface tunnel 2
 ip unnumbered loopback1
```

```
 tunnel destination 10.0.1.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute
 tunnel mpls traffic-eng path-option 1 explicit name P1-P2
ip explicit-path name P1-P2 enable
 next-address 10.0.0.1
 next-address 10.0.0.2
 next-address 10.0.1.2
```

Število vrstic je statično in raste le z dolžino poti. V prikazanem primeru nam usmerjevalnik PE1 ustvari tunelsko povezavo do usmerjevalnika PE2 z eksplicitno definirano potjo.

4.4 Primerjava obeh konfiguracij

Na prvi pogled se zdi, da je drugi način konfiguracije enostavnejši. Vendar je treba upoštevati naslednje:

- Konfiguracija usmerjevalnikov P v prvem primeru je splošna, ni vezana na število in usmeritev tunelskih povezav, velikost pa je odvisna od števila fizičnih vmesnikov, vključenih v hrbtenično omrežje. Konfiguracija se ne spreminja v odvisnosti od tunelskih povezav.

- Konfiguracija usmerjevalnikov P v drugem primeru je konfiguracija, neposredno vezana na usmerjanje tunelskih povezav. Za vsako tunelsko povezavo potrebujemo dodatno konfiguracijo na vseh usmerjevalnikih P, preko katerih je tunelska povezava speljana.

Število dodatnih konfiguracijskih vrstic raste linearno s številom tunelskih povezav. Upravljanje s tako konfiguracijo lahko hitro pripelje do napak. Vpeljava nove tunelske povezave zahteva novo konfiguracijo povratnega vmesnika na usmerjevalnikih PE ter statične usmeritve prometa do tega vmesnika na tranzitnih usmerjevalnikih P. Premik poti tunelske povezave zahteva rekonfiguracijo starih in novih tranzitnih usmerjevalnikov P.

V tabeli 4.4 lahko vidimo število potrebnih vrstic na posameznem tipu usmerjevalnika v odvisnosti od parametrov, kot so število tunelskih povezav na usmerjevalniku (P), število fizičnih vmesnikov na usmerjevalniku (I) ter povprečna dolžina tunelske povezave (L).

Pri pristopu s tuneli MPLS TE količina konfiguracije narašča s številom vmesnikov. V splošnem število vmesnikov na usmerjevalnikih ne bo naraščalo

Tabela 4.4: Potrebno število vrstic konfiguracije.

tip usmerjevalnika	Klasični	MPLS TE
vrstic na tranzitnem P	$2P$	$3 + 2I$
vrstic na netranzitnem P	0	$3 + 2I$
vrstic na PE	$7P$	$7(P + L) + 3 + 2I$

z večanjem omrežja. Precej hitreje lahko narašča število tunelskih povezav, če želimo več kontrole nad prometnimi tokovi. Iz formul v tabeli 4.4 je razvidno, da pri obeh pristopih število vrstic na usmerjevalnikih PE narašča predvsem s številom povezav. Klasični pristop ima pri ocenjevanju kompleksnosti s številom dodatnih vrstic celo nekaj prednosti, razen pri tranzitnih usmerjevalnikih P. Kar ni očitno iz tabele, je razpršenost informacije o posamezni tunelski povezavi. Pri klasičnem pristopu je ta informacija razpršena po vseh tranzitnih usmerjevalnikih P, pri pristopu s tuneli MPLS TE pa je locirana le na eni sami lokaciji. Spreminjanje fiksiranih poti je tako mnogo lažje pri pristopu s tuneli MPLS TE.

4.5 Primer implementacije

V tem podpoglavju sta prikazani dve implementaciji kot primer delujoče kontrole toka prometa. Najprej je prikazano stanje brez prometnega urejanja, sledi rešitev s klasičnimi tuneli in nazadnje še rešitev s tuneli MPLS TE.

4.5.1 Izpisi začetnega stanja

Sledijo izpisi ukazov ob začetnem stanju. V tem stanju promet sledi optimalnim potem, ki jih je izračunal usmerjevalni protokol.

Ukaz “show ip route” nam izpiše izhodno pot za vse ali za navedeno ciljno podomrežje. Tu je uporabljen za prikaz izhodnih poti do dveh ciljnih usmerjevalnikov.

```
pe1#sh ip route 10.0.1.2
Routing entry for 10.0.1.2/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.0.113.3 on FastEthernet0/1, 00:40:16 ago
  Routing Descriptor Blocks:
```

```
  * 10.0.113.3, from 10.0.1.2, 00:40:16 ago, via FastEthernet0/1
      Route metric is 3, traffic share count is 1
```

V tem izpisu vidimo, da je izhodni vmesnik za cilj 10.0.1.2 (usmerjevalnik PE2) FastEthernet 0/1.

```
pe1#sh ip route 10.0.1.3
Routing entry for 10.0.1.3/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.0.113.3 on FastEthernet0/1, 00:01:39 ago
  Routing Descriptor Blocks:
  * 10.0.113.3, from 10.0.1.3, 00:01:39 ago, via FastEthernet0/1
      Route metric is 3, traffic share count is 1
```

Iz zgornjega izpisa je očitno, da je tudi za cilj 10.0.1.3 (usmerjevalnik PE3) izhodni vmesnik FastEthernet 0/1. Temu se želimo izogniti. Ukaz "traceroute" pokaže celotno pot do ciljne naprave. Uporablja pakete UDP z nizkimi vrednostimi TTL ter se zanaša na dogovor, da usmerjevalnik, ki dobi paket s poljem TTL, enakim 1, odgovori, da paketa ne more poslati dalje. S tem principom lahko vidimo pot do destinacije, pod pogojem, da se vsi usmerjevalniki ravnajo po predpisanih pravilih ter da na poti nihče ne filtrira prometa. V našem testnem okolju je temu zadoščeno.

Naslednji izpis ukaza "traceroute" nam prikazuje pot od PE1 do PE2:

```
pe1#traceroute 10.0.1.2

Type escape sequence to abort.
Tracing the route to 10.0.1.2

  1 10.0.113.3 44 msec 44 msec 4 msec
  2 10.0.123.12 8 msec *  100 msec
```

Pot, ki jo usmerjevalniki poročajo, je od PE1 do P3 ter od P3 do PE2.

Naslednji izpis nam kaže pot od PE1 do PE3:

```
pe1#traceroute 10.0.1.3

Type escape sequence to abort.
```

```
Tracing the route to 10.0.1.3

  1 10.0.113.3 96 msec 68 msec 4 msec
  2 10.0.133.13 36 msec *  68 msec
```

Pot, ki jo usmerjevalniki poročajo, je od PE1 do P3 ter od P3 do PE3.
Na usmerjevalniku PE2 dodamo 10.0.2.2 na virtualni vmesnik:

```
pe1#trace 10.0.2.2

Type escape sequence to abort.
Tracing the route to 10.0.2.2

  1 10.0.113.3 48 msec 48 msec 4 msec
  2 10.0.123.12 40 msec *  36 msec
```

Pot do njega je identična kot za 10.0.1.2.

4.5.2 Rešitev s klasičnimi tuneli

Dodatna konfiguracija za usmeritev tunela:

```
PE1  ip route 10.0.2.2 255.255.255.255 10.0.0.1
P1   ip route 10.0.2.2 255.255.255.255 10.0.0.2
P2   ip route 10.0.2.1 255.255.255.255 10.0.0.1
PE2  ip route 10.0.2.1 255.255.255.255 10.0.0.2
```

Na vseh vmesnih usmerjevalnikih, preko katerih želimo speljati promet, dodamo statične usmeritve. Dvojne statične usmeritve na P1 in P2 niso potrebne, saj je optimalna pot do PE1 oziroma PE2 enaka tisti, ki jo želimo.

Konfiguracija tunelskih vmesnikov je naslednja:

```
interface Tunnel1
 ip address 10.0.201.2 255.255.255.252
 bandwidth 100000
 tunnel source Loopback1
 tunnel destination 10.0.2.1
 tunnel mode ipip
end
```

```
interface Tunnel1
 ip address 10.0.201.1 255.255.255.252
 bandwidth 100000
 tunnel source Loopback1
 tunnel destination 10.0.2.2
 tunnel mode ipip
end
```

Tunel "pripnemo" na virtualne vmesnike, za katere smo določili statične usmeritve. Način enkapsulacije je nastavljen na "ipip", ki zagotavlja najmanjše dodatno breme za vsak paket (20 oktetov). Kapaciteta tunelske povezave mora biti identična ostalim fizičnim povezavam, da jim lahko konkurira v izboru. Privzeta kapaciteta je tako majhna (8 kb/s), da usmerjevalni protokol še vedno izbere pot po fizičnih povezavah. Rezultat je naslednji:

```
pe1#sh ip route 10.0.1.2
Routing entry for 10.0.1.2/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 10.0.201.2 on Tunnel1, 00:01:23 ago
  Routing Descriptor Blocks:
  * 10.0.201.2, from 10.0.1.2, 00:01:23 ago, via Tunnel1
      Route metric is 2, traffic share count is 1
```

Izhodni vmesnik proti 10.0.1.2 (PE2) je zdaj Tunnel1, izbran pa je dinamično, tako kot prej.

```
pe1#sh ip route 10.0.1.3
Routing entry for 10.0.1.3/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.0.113.3 on FastEthernet0/1, 00:11:53 ago
  Routing Descriptor Blocks:
  * 10.0.113.3, from 10.0.1.3, 00:11:53 ago, via FastEthernet0/1
      Route metric is 3, traffic share count is 1
```

Izhodni vmesnik proti PE3 je še vedno fizični vmesnik. S tem smo dosegli, da promet proti PE2 obremenjuje druge fizične povezave kot promet proti PE3.

Preveriti je treba še, katero pot izbere usmerjevalni protokol za usmeritev tunelske povezave:

```
pe1#trace 10.0.2.2

Type escape sequence to abort.
Tracing the route to 10.0.2.2

  1 10.0.111.1 112 msec 48 msec 4 msec
  2 10.0.12.2 40 msec 56 msec 16 msec
  3 10.0.122.12 40 msec *  100 msec
```

Promet gre od PE1 do P1, do P2, nato do PE2.

Promet, ki se pošlje v tunelsko povezavo, ne vidi dejanske poti:

```
pe1#traceroute 10.0.1.2

Type escape sequence to abort.
Tracing the route to 10.0.1.2

  1 10.0.201.2 108 msec *  120 msec
```

Vidi le navidezno povezavo.

4.5.3 Rešitev s tuneli MPLS

Konfiguracija na usmerjevalniku PE1:

```
interface tunnel 2
 ip unnumbered loopback1
 tunnel destination 10.0.1.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute
 tunnel mpls traffic-eng path-option 1 explicit name P1-P2
ip explicit-path name P1-P2 enable
 next-address 10.0.0.1
 next-address 10.0.0.2
 next-address 10.0.1.2
```

Konfiguracija na usmerjevalniku PE2:

```
interface tunnel 2
 ip unnumbered loopback1
 tunnel destination 10.0.1.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute
 tunnel mpls traffic-eng path-option 1 explicit name P1-P2
ip explicit-path name P2-P1 enable
 next-address 10.0.0.2
 next-address 10.0.0.1
 next-address 10.0.1.1
```

Tunel je pripet na prvi virtualni vmesnik in ne na drugega kot pri klasičnih tunelih. Razlog je v tem, da nam tu ni treba drugače usmeriti prometa do virtualnega vmesnika, ker je pot tunelskega prometa eksplicitno določena z definirano potjo. Rezultat je naslednji:

```
pe1#sh ip route 10.0.1.2
Routing entry for 10.0.1.2/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.0.1.2 on Tunnel2, 00:00:23 ago
  Routing Descriptor Blocks:
  * 10.0.1.2, from 10.0.1.2, 00:00:23 ago, via Tunnel2
      Route metric is 3, traffic share count is 1
```

Pot do usmerjevalnika PE2 gre preko vmesnika Tunnel2.

```
pe1#sh ip route 10.0.1.3
Routing entry for 10.0.1.3/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.0.113.3 on FastEthernet0/1, 00:00:52 ago
  Routing Descriptor Blocks:
  * 10.0.113.3, from 10.0.1.3, 00:00:52 ago, via FastEthernet0/1
      Route metric is 3, traffic share count is 1
```

Pot do usmerjevalnika PE3 ostaja preko vmesnika FastEthernet0/1.

```
pe1#trace 10.0.1.2

Type escape sequence to abort.
Tracing the route to 10.0.1.2

  1 10.0.111.1 [MPLS: Label 17 Exp 0] 72 msec 104 msec 16 msec
  2 10.0.12.2 [MPLS: Label 17 Exp 0] 136 msec 56 msec 56 msec
  3 10.0.122.12 44 msec *  132 msec
```

Pot od PE1 do 10.0.1.2 (PE2) gre sedaj preko povezav PE1−P1, potem P1−P2 in nazadnje P2−PE2. Pot je enaka poti klasičnega tunela. V primeru tunelov MPLS tuneliran promet vidi dejansko pot, prav tako pa je možno videti labele.

Poglavje 5

Sklepne ugotovitve

Primerjava pristopov je pokazala, da manipulacija metrike, kot je predstavljena v poglavju 2, ni vedno smiselna ali možna rešitev. Kompleksnost nastavitev raste preko obvladljivih meja, če dodajamo omejitve za pot prometa. Taka rešitev je popolnoma sprejemljiva v zelo majhnih omrežjih, kjer prednost naprednejših rešitev ne pride do izraza zaradi preprostosti problema.

Deljenje prometa med dvema točkama v omrežju preko različnih povezav se je v testih pokazalo kot performančno neustrezno. Izkaže se, da je v primeru, ko imamo na voljo več vzporednih povezav, performančno bolje uporabiti eno kot pa deliti promet iste seje. V praksi se administratorji omrežij izogibajo deljenju prometa, hkrati pa tudi omrežne naprave vsebujejo mehanizme, ki jim preprečujejo, da bi poslali promet ene seje po različnih povezavah. Na primer, omrežna stikala omogočajo združevanje več fizičnih povezav Ethernet v eno logično povezavo. Vendar stikalo vedno pošilja promet ene seje po eni sami fizični povezavi. Le v primeru, da imamo mnogo naprav, ki govorijo med sabo, se promet porazdeli enakomerno po vseh povezavah.

Preusmeritev prometa s statičnim usmerjanjem je neprimerna rešitev z vidika zanesljivosti. Če pride v omrežju do odpovedi neke povezave, statično usmerjanje ne zna izkoristiti nadomestnih povezav. Na ta način lahko povzročimo, da je omrežje sicer bolje izkoriščeno, v primeru napak pa se ne more dinamično odzivati in preusmeriti prometa.

Kreiranje nove tunelske povezave, ki je statično fiksirana v omrežje in ima boljšo metriko, se je pokazalo kot najboljša izbira. S tem pristopom pridobimo prednosti statične določitve poti prometa skozi omrežje, ki sama po sebi ni optimalna, lahko pa na ta način obremenimo sicer neuporabljene povezave v omrežju. Hkrati pa obdržimo možnost dinamičnega prilagajanja na dogajanje v omrežju. Če pride do odpovedi povezave in se statično definirana pot skozi

omrežje prekine, bo dinamično usmerjanje zaznalo, da je tunelska povezava neuporabna in preusmerilo promet nekam drugam, dokler obstaja alternativna pot.

Primerjal sem dva pristopa usmerjanja tunelov. Prvi je bil klasično usmerjanje klasičnih (GRE ali IP v IP) tunelov. Drugi je bil uporaba tunelov MPLS TE z definirano potjo skozi omrežje na začetnem usmerjevalniku tunela. Uporabna sta oba pristopa. Izbira je odvisna od omrežja, v katerem želimo izvajati prometno načrtovanje. Če je v tem omrežju tehnologija MPLS že uporabljena in so prisotni vsi potrebni pogoji za implementacijo tehnologije MPLS TE (uporaba notranjega usmerjanja tipa "link-state", eno samo območje usmerjanja (angl. *routing area*), podpora na usmerjevalnikih) ter pričakujemo več kot en sam tunel v omrežju, je precej bolj smiselno uporabiti pristop MPLS TE. Če teh pogojev nimamo ali če osebje nima znanja, kako uporabiti pristop s prometnim načrtovanjem, je smiselno uporabiti pristop s klasičnimi tuneli. Če pogledamo s stališča performans, kompleksnosti konfiguracije ter skalabilnosti, vse kaže, da je pravi pristop uporaba prometnega načrtovanja. Ob dodajanju tunelov v omrežje je rast konfiguracije manjša, preglednost pa večja.

Trenutne implementacije MPLS TE na platformah Cisco podpirajo dinamične in statične tunele MPLS TE. Pri statičnih definiramo pot tunela, pri dinamičnih pa le omejitve kapacitet. Prednost statičnih tunelskih povezav je možnost, da promet res usmerimo tja, kamor želimo. Lahko natančno definiramo pot prometa za vsak skok posebej in tako izberemo tiste povezave, ki so najmanj obremenjene. Slabost tega pristopa je, da so tunelske povezave popolnoma statične in nefleksibilne. Ob prekinitvi tranzitne povezave na poti je tunelska povezava neuporabna in dinamično usmerjanje bo izbralo drugo pot. Če je bila to edina tunelska povezava do našega cilja, se bo promet preusmeril na fizične povezave. S tem ni nič narobe, le ubral bo isto pot kot ves ostali promet in s tem povečal možnost, da se bodo optimalne poti spet zasičile, alternativne pa ostale prazne. Z uporabo dinamičnih poti lahko definiramo zahteve po kapacitetah tunelske povezave, še prej pa moramo definirati kapacitete fizičnih povezav. Protokol RSVP poskrbi za alokacijo zahtevanih kapacitet. S tem sicer zagotovimo, da bo vsaka tunelska povezava dobila na voljo toliko dejanskih kapacitet, kot jih zahtevamo, ne moremo pa zagotoviti natančne poti skozi omrežje. Prav tako pri množici tunelskih povezav v omrežju usmerjevalniki ne uporabljajo učinkovitih algoritmov za izračun optimalnega izkoriščenja vseh povezav v grafu. Časovna zahtevnost takih algoritmov bi bila neprimerna za hitro odzivanje na spremembe v omrežju. Kombinacija obeh pristopov z vnaprej izračunanimi alternativnimi potmi bi bila vsekakor možna rešitev. Na ta način bi lahko del poti definirali vnaprej, na podlagi rezultatov počasnih

algoritmov. Za druge dele poti pa bi lahko izbrali enostavnejše dinamično prilagajanje, ki bi se lahko hitro odzvalo na odpovedi povezav v omrežju.

Slike

Tabele

Literatura

[1] Cisco Systems, "Configuring a Load-Balancing Scheme for Cisco Express Forwarding Traffic," julij 2008. Dostopno na: http://www.cisco.com/en/US/docs/ios/ipswitch/configuration/guide/-cef_load_balancng.html [2.5.2010]

[2] Cisco Systems, "How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP?," junij 2009. Dostopno na: http://www.cisco.com/en/US/tech/tk365/-technologies_tech_note09186a008009437d.shtml [2.5.2010]

[3] Cisco Systems, "Policy-Based Routing". Dostopno na: http://www.cisco.com/en/US/products/ps6599/-products_white_paper09186a00800a4409.shtml [2.5.2010]

[4] Cisco Systems, "MPLS Traffic Engineering (TE) – Configurable Path Calculation Metric for Tunnels," februar 2008. Dostopno na: http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/-fsmetric.html [2.5.2010]

[5] Information Sciences Institute, USC, "Internet Protocol,"RFC 791, september 1981. Dostopno na: http://tools.ietf.org/html/rfc791 [2.5.2010]

[6] G. Malkin, "RIP Version 2," RFC 2453, november 1998. Dostopno na: http://tools.ietf.org/html/rfc2453 [2.5.2010]

[7] , E. Rosen, Y. Rekhter, "BGP/MPLS VPNs," RFC 2547, marec 1999 . Dostopno na: http://tools.ietf.org/html/rfc2547 [2.5.2010]

[8] JP. Vasseur, Ed., Y. Ikejiri, R. Zhang, "Reoptimization of Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Loosely Routed Label

Switched Path (LSP),” RFC 4736, november 2006. Dostopno na: http://tools.ietf.org/html/rfc4736 [2.5.2010]