





RIP usmjernici šalju svoje kompletne tablice usmjeravanja susjedima, i to svakih 30 sekundi, osim za one rute koje su obuhvaćene pravilom podijeljenog obzorja (engl. Split Horizon). Split Horizon je pravilo koje sprečava pojavu petlji kod Distance Vector protokola usmjeravanja. Prema tom pravilu, informacija o ruti se nikad ne šalje natrag kroz sučelje na koje je ažuriranje došlo.

Korištenje RIP protokola se omogućuje naredbom *router rip* u globalnom konfiguracijskom modu usmjernika. Naredba *network* se koristi za određivanje sučelja usmjerivača koja će biti omogućena za RIP sa classful mrežnim adresama za svaku izravno spojenu mrežu. Ta naredba osposobljava sučelja za primanje i slanje RIP ažuriranja.

Naredba *debug ip rip* se može koristiti za prikaz RIP ažuriranja koja su poslana i primljena na usmjerniku. Da bi se spriječilo slanje RIP ažuriranja kroz određeno sučelje, za LAN gdje nema drugih usmjernika, koristi se naredba *passive-interface*.

U tablici usmjeravanja su RIP unosi prikazani slovom R i imaju administrativnu udaljenost 120. Zadane (engl. default) rute se propagiraju korištenjem naredbe *default-information originate* u konfiguracijskom modu usmjernika.

Pošto je RIPv1 *classful* protokol usmjeravanja, ne podržava nesusedne (engl. Discontiguous) mreže i VLSM (engl. Variable Length Subnet Masks), te maska podmreže nije uključena u usmjernička ažuriranja. Kada usmjernik primi RIPv1 ažuriranje, RIP mora odrediti masku podmreže te rute. Ako ruta pripada istoj glavnoj classful mreži kao i ažuriranje, RIPv1 uzima masku podmreže primajućeg sučelja. Ako ruta pripada drugoj glavnoj classful mreži, RIPv1 primjenjuje zadanu classful masku. Naredbom *show ip protocols* mogu se prikazati informacije o bilo kojem protokolu usmjeravanja koji je omogućen na usmjerniku. U slučaju RIP-a, ova naredba prikazuje informacije brojača, status automatske sumarizacije, omogućene mreže na usmjerniku i dr.

### 3.2. CIDR i VLSM

CIDR (engl. Classless Inter-Domain Routing) je uveden 1993. godine zamijenivši prethodnu generaciju IP adresne sintakse classful mreža. CIDR dopušta učinkovitije korištenje IPv4 adresnog prostora i agregaciju prefiksa poznatu kao sumarizacija rute ili supermrežiranje. S CIDR-om klase adresa (klasa A, klasa B, klasa C) postaju besmislene. Mrežne adrese više nisu određene vrijednostima prvog okteta, ali je dodijeljena dužina prefiksa ili maska podmreže. Adresni prostor i broj korisnika na mreži sada može biti određen prefiksom koji ovisi o broju korisnika potrebnih za određenu mrežu.

CIDR koristi VLSM kako bi dodijelio IP adrese podmrežama po potrebi, a ne po klasama. VLSM omogućuje da podmreža može biti dodatno podijeljena u još manje podmreže [6]. VLSM je zapravo subnetiranje podmreža. Propagiranje CIDR supermreža ili VLSM podmreža zahtijeva classless (besklasni) protokol usmjeravanja koji u ažuriranjima zajedno s mrežnom adresom uključuje masku podmreže.

Utvrđivanje sumirizirane rute i maske podmreže za grupu mreža se može izvesti u tri jednostavna koraka. Prvi je da se mreže pretvore u binarni format, drugi je da se s lijeva na desno zbroje bitovi koji se podudaraju tako da se dobije duljina prefiksa ili maska podmreže sumirizirane rute. Zadnji korak je kopiranje bitova koji se podudaraju i dodavanje nula na ostatak adrese.

### 3.3. RIPv2

Ovo je classless Distance Vector protokol usmjeravanja. Zbog toga u ažuriranjima zajedno s adresama mreža uključuje masku podmreže. Podržava CIDR supermreže, VLSM i nesusedne mreže.

Classful protokoli poput RIPv1 ne mogu podržavati nesusedne mreže jer automatski sumiriziraju na glavnim mrežnim granicama. Usmjernik koji prima ažuriranje od više usmjernika koji oglašavaju istu classful sumiriranu rutu, ne može otkriti koje podmreže pripadaju kojoj sumiriranoj ruti. Ova nemogućnost dovodi do neočekivanih rezultata i pogrešnog usmjeravanja paketa.

Na usmjerniku je inicijalno zadana verzija RIPv1 pa se naredbom *version 2* mijenja verzija u RIPv2. Slično kao i RIPv1, RIPv2 automatski sumirizira na glavnim mrežnim granicama, no kod RIPv2 se automatska sumarizacija može isključiti naredbom *no auto-summary*. Automatska sumarizacija mora biti onemogućena kako bi se mogle podržati nesusedne mreže. Maska podmreže je uključena uz svaku mrežnu adresu u ažuriranjima, a naredbom *debug ip rip* se mogu vidjeti ažuriranja i maske podmreže poslana zajedno s adresama mreža. Sve ostale naredbe kod konfiguracije protokola su identične kao i kod RIPv1.

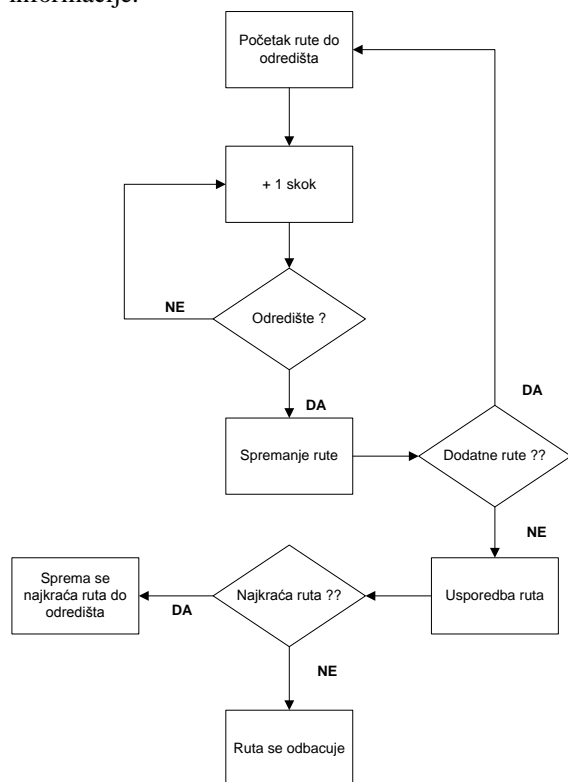
### 3.4. Bellman-Ford algoritam

Nazvan je po američkim matematičarima Richardu Bellmanu i Lesteru Fordu mlađem. Bellman-Ford algoritam pronalazi najkraće putove od početne točke s ograničenjem na jedan link (vezu), zatim na dva linka i tako dalje. Prvi korak je traženje točaka udaljenih za jedan skok, zatim se nalaze točke udaljene za dva skoka itd. Tablica se izrađuje ovisno o broju linkova. Informacije se izmjenjuju sa susjednim usmjernicima, a svaki podatak sadrži put za slanje paketa do svake destinacije u mreži i udaljenost ili vrijeme za slanje na tom putu (cijena). Proces izmjenjivanja informacija se događa i ako se ne događaju promjene u susjedstvu. Ako graf sadrži negativni ciklus, tj. ciklus čija suma rubova ima negativnu vrijednost, onda nema najbolji put i to zbog toga jer svaka ruta može biti bolja od još jednog kretanja kroz negativni ciklus. U takvim slučajevima Bellman-Ford algoritam može otkriti negativne cikluse i ukazati na njihovo postojanje, ali ne može stvoriti odgovor s pravim najkraćim putem. Praktična primjena ovog algoritma je u RIP protokolu usmjeravanja [7]. Sastoji se od sljedećih koraka:

1. Svaki usmjernik izračunava udaljenost između sebe i svih ostalih čvorova unutar autonomnog sustava i te informacije pohranjuje u svoju tablicu.

2. Svaki usmjernik šalje svoju tablicu svim susjednim usmjernicima.
3. Kada usmjernik primi tablice udaljenosti od svojih susjeda, izračunava najkraće puteve do svih ostalih usmjernika i ažurira vlastitu tablicu da bi zabilježio svaku promjenu.

Nedostatak je što se ažuriranja ne događaju trenutno jer se šire od točke do točke kroz mrežu. Zbog toga se ponekad u tablici usmjerenja mogu pojaviti krive informacije.



Slika 2. Dijagram toka Bellman-Ford algoritma

## 4. EIGRP

EIGRP je classless Distance Vector protokol usmjerenja. U tablici usmjerenja za rute koristi slovo D. Zadana administrativna udaljenost mu je 90 za interne rute i 170 za rute primljene iz vanjskog izvora.

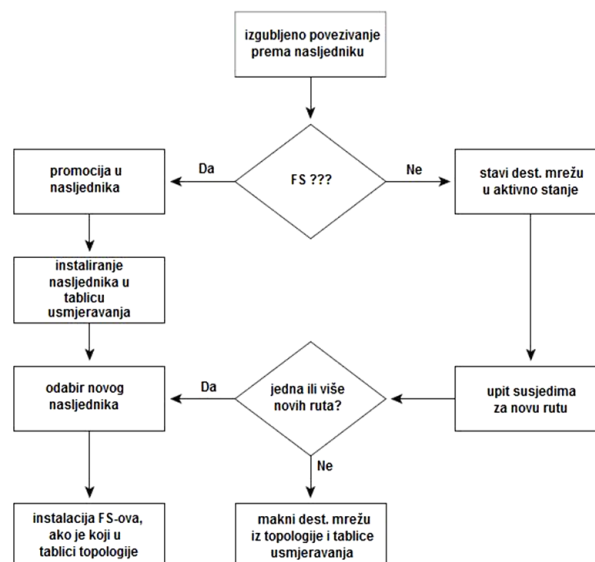
Za dostavu EIGRP paketa koristi RTP (engl. Reliable Transport Protocol) kao protokol na transportnom sloju. Koristi pouzdanu isporuku EIGRP ažuriranja, upite i odgovore te nepouzdanu isporuku za EIGRP pozdrave i potvrde (engl. Hellos and Acknowledgments). Pouzdani RTP znači da se mora vratiti EIGRP potvrda. Prije nego se ažuriranje pošalje, usmjernik otkriva svoje susjedstvo i to s EIGRP *hello* paketima. Na većini mreža *hello* paketi se šalju svakih 5 sekundi. Vrijeme zadržavanja (engl. Hold Time) je tri puta duže od *hello*-a, tj. 15 sekundi. Provjera uspostavljenih veza sa susjednim usmjernicima je moguća naredbom *show ip eigrp neighbors*.

EIGRP ne šalje periodička ažuriranja poput RIP-a nego šalje djelomična ili ograničena ažuriranja koja uključuju samo promjene ruta i to samo onim usmjernicima kojima je potrebna ta informacija. Kod određivanje najboljeg puta koristi složenu metriku sastavljenu od propusnosti, kašnjenja, pouzdanosti i

opterećenja. Inicijalno je definirano da se koristi samo propusnost i kašnjenje. Zadani izračun je najmanja propusnost plus zbroj svih kašnjenja od izlaznog sučelja usmjerniča do određene mreže.

### 4.1. DUAL algoritam

U središtu EIGRP-a je DUAL (engl. Diffusing Update Algorithm). Kako bi se odredio najbolji put i potencijalni zamjenski putevi, koristi se DUAL FSM (engl. Finite State Machine). Nasljednik je susjedni usmjernik koji se koristi za prosljeđivanje paketa koristeći rutu s najmanje troškova do određene mreže. Izvodljiva udaljenost (engl. Feasible Distance, FD) je najmanja izračunata metrika za dohvaćanje određene mreže preko nasljednika. Izvodljivi nasljednik (engl. Feasible Successor, FS) je susjed koji ima do iste mreže kao i nasljednik rezervni put bez petlji, a također zadovoljava i uvjete izvodljivosti (engl. Feasible Condition, FC). FC je zadan kada je susjedova izvještana udaljenost (engl. Reported Distance, RD) do mreže manja od FD-a do iste destinacijske mreže. Jednostavno, RD je susjedov FD do destinacijske mreže.



Slika 3. Dijagram toka FSM DUAL algoritma [8]

EIGRP se konfigurira naredbom *router eigrp* u kojoj treba odrediti i broj autonomnog sustava. Vrijednost autonomnog sustava je zapravo process-id i mora biti ista na svim usmjernicima u EIGRP domeni usmjerenja. Naredba *network* je slična onoj koja se koristi kod RIP-a, ali se uz adresu mreže stavlja wildcard maska ili obrnuta maska. Za propagaciju statičkih ruta se koristi naredba *redistribute static*.

## 5. OSPF

OSPF je classless Link State protokol usmjerenja. Ima zadanu administrativnu udaljenost 110, a u tablici usmjerenja je označen slovom O. Aktivira se naredbom *router ospf* uz koju se još dodaje process-id i to u globalnom konfiguracijskom modu. Process-id je lokalno značajan, što znači da ne mora odgovarati

drugim OSPF usmjernicima kako bi se uspostavilo susjedstvo s njima.

Naredba *network* ima istu funkciju kao i kod ostalih IGP protokola usmjeravanja, ali s malo drugačijom sintaksom.

```
Router(config-router)#network mrežna-adresa  
wildcard-mask area area-id
```

*Wildcard* maska je inverzija maske podmreže, a *area-id* bi trebao biti postavljen na 0.

Za uspostavljanje susjedstva koriste se OSPF *hello* paketi koji se šalju svakih 10 sekundi na višepristupnim i point-to-point (od točke do točke) segmentima i 30 sekundi na NBMA segmentima (engl. Non-Broadcast Multiaccess). Svaki usmjernik čeka određeno vrijeme (engl. *dead interval*) prije nego prekine susjedstvo sa susjedom. *Dead interval* je četiri puta duži od *hello* intervala.

Da bi usmjernici uspostavili susjedstvo, njihovi *hello* i *dead* intervali, te vrsta mreže i maska podmreže, moraju odgovarati. Za provjeru susjedstva koristi se naredba *show ip ospf neighbors*.

Router-ID jedinstveno identificira svaki usmjernik u domeni, a dobiva se na temelju tri kriterija:

1. koristi se IP adresa konfigurirana naredbom *router-id*
2. ako *router-id* nije konfiguriran, usmjernik bira najvišu IP adresu bilo kojeg od svojih loopback sučelja
3. ako nema loopback sučelja, usmjernik bira najvišu aktivnu IP adresu od bilo kojeg svog fizičkog sučelja

Loopback sučelje je softver koji se koristi kao simulacija fizičkog sučelja. Poput drugih sučelja, može imati svoju IP adresu. U laboratorijskom okruženju loopback sučelja su korisna u kreiranju dodatnih mreža bez potrebe za fizičkim sučeljima na usmjerniku.

Za određivanje vrijednosti cijena u metrici koriste se propusnosti izlaznih sučelja usmjerivača prema određenoj mreži. Višepristupne mreže mogu stvoriti različite izazove što se tiče poplava (engl. *flood*) s LSA-ovima (engl. *Link State Advertisement*), uključujući i stvaranje višestrukih adjacencija (povezanosti sa susjedima). Dolazi do opsežnih poplava LSA-ova pa OSPF bira DR (engl. *Designated Router*) koji je točka za prikupljanje i distribuciju LSA-ova poslanih i primljenih u višepristupnoj mreži. BDR (engl. *Backup Designated Router*) je odabran da preuzme ulogu DR-a u slučaju da DR padne. Svi ostali usmjernici su poznati kao DRother. Svi usmjernici šalju svoje LSA-ove prema DR-u koji zatim tim informacijama „poplavi“ sve ostale usmjernike u višepristupnoj mreži.

Usmjernik koji ima najviši *router-ID* je DR, a usmjernik sa sljedećim najvišim *router-ID*-om je BDR. To se može izmijeniti naredbom *ip ospf priority* u modu sučelja. IP OSPF prioritet je zadan na 1 na svim višepristupnim sučeljima. Ako je usmjernik konfiguriran s novom vrijednošću prioriteta, usmjernik s najvišim prioritetom je DR, a sljedeći najviši je BDR. Vrijednost prioriteta 0 znači da usmjernik ne može postati ni DR ni BDR.

Zadana ruta se propagira slično kao i kod RIP-a. Za propagaciju statičke zadane rute koristi se naredba *default-information originate*. Važna naredba za provjeru

konfiguracijskih informacija, *process ID*-a, *router ID*-a i mreža koje usmjernik oglašava, je *show ip protocol*.

Kod mrežnih protokola, ponajviše kod OSPF i IS-IS, naširoko se koristi algoritam traženja najkraćeg puta od početnog vrha, izvora, do krajnjeg vrha, cilja, poznatiji pod nazivom Dijkstrin algoritam [9]. Algoritam je otkrio nizozemski računalni stručnjak Edsger Dijkstra 1959. godine. Osim algoritma traženja najkraćeg puta, Dijkstrin doprinos u računarstvu očituje se i u razvoju programskog jezika ALGOL, razvoju tzv. semafora za nadzor računalnih procesa te u razvoju računalnih operativnih sustava. U sljedećem poglavlju dane su osnove teorije grafova te je opisana procedura Dijkstrinog algoritma.

## 6. TEŽINSKI GRAF. DIJKSTRIN ALGORITAM

Graf  $G$  je uređeni par  $G = (V, E)$  koji se sastoji od nepraznog skupa  $V = V(G)$  čiji su elementi vrhovi od  $G$  i skupa  $E = E(G)$  čiji su elementi bridovi od  $G$ . Šetnja u grafu  $G$  je naizmjeničan niz vrhova i bridova. Ako su u šetnji svi vrhovi različiti, šetnja se naziva put. Graf je povezan ako postoji put između svaka dva vrha grafa. Funkcija  $w: E(G) \rightarrow R^+$  koja svakom bridu  $e \in E(G)$  pridružuje broj  $w(e) \in R^+$  naziva se težinska funkcija, a broj  $w(e)$  težina brida  $e \in E(G)$ . Bridovima su dakle pridružene težine koje predstavljaju npr. duljinu ceste u kilometrima, vrijeme izraženo u satima, cestarinu u kunama i sl. Graf  $G$  čijim su bridovima pridružene težine naziva se težinski graf. Za podgraf  $H$  grafa  $G$  broj  $w(H) = \sum_{e \in E(H)} w(e)$  je

težina podgraфа  $H$ . Dakle, težina podgraфа  $H$  grafa  $G$  je suma težina svih bridova u grafu. Udaljenost  $d(u, v)$  dva vrha  $u, v \in V(G)$  u težinskom grafu je duljina najkraćeg puta između vrhova  $u$  i  $v$ , a najkraći put između vrhova  $u$  i  $v$  u težinskom grafu smatra se put najmanje težine. Detaljnije o teoriji grafova vidi u [10]. Problem najkraćeg puta jedan je od najvažnijih algoritamskih problema sa širokom praktičnom primjenom. Problem najkraćeg puta u težinskom grafu svodi se na traženje puta između dva vrha s najmanjom težinom. Slijedom spomenutog, jedan od najpoznatijih algoritama za traženje putova najmanje težine u težinskom grafu s pozitivnim težinama je Dijkstrin algoritam. Algoritam otkriva najkraći put iz određenog vrha, označenog s 1, do svih drugih vrhova 2, 3, ...,  $n$  u  $G$ . Neka je  $D_j$  duljina najkraćeg puta između vrhova 1 i  $j$  u  $G$ . U svakom koraku svaki vrh  $v$  dobiva ili trajnu oznaku, oznaku *TO* te se s  $D_v$  označava najkraći put od vrha 1 do vrha  $v$ , ili vrh dobiva privremenu oznaku, oznaku *PO* te se s  $\tilde{D}_v$  označava gornja međa duljine najkraćeg puta od vrha 1 do vrha  $v$ . Neka je

$S_{TO}$  skup vrhova s trajnom oznakom, a  $S_{PO}$  skup vrhova s privremenom oznakom.

Za povezani graf  $G = (V, E)$  s vrhovima  $1, 2, \dots, n$  i udaljenostima  $d(i, j) > 0$  od vrha  $i$  do vrha  $j$ , procedura izračunavanja duljine najkraćeg puta od vrha 1 do ostalih vrhova  $2, 3, \dots, n$  u  $G$  je sljedeća [11]:

ULAZ:  $n$  vrhova, bridovi  $(i, j)$  i udaljenosti  $d(i, j)$

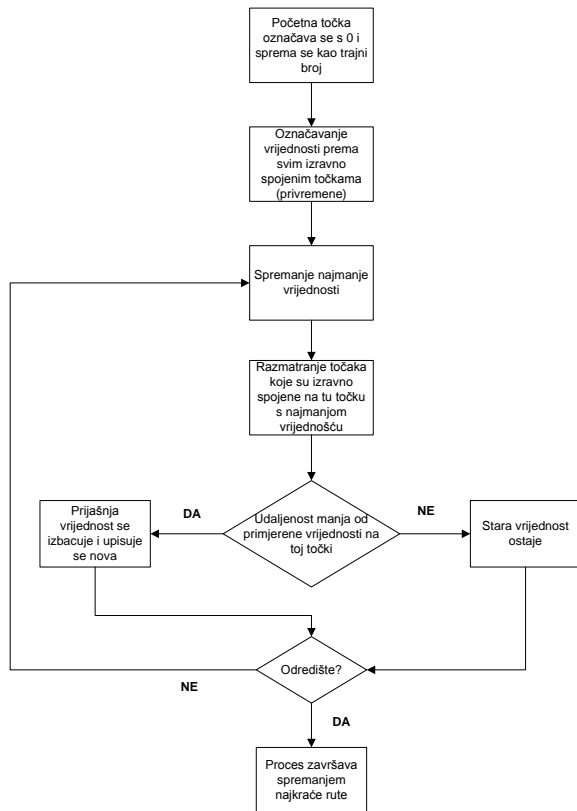
IZLAZ: duljine  $D_j$  najkraćih putova od vrha 1 do vrha  $j$ ,  $j = 2, 3, \dots, n$

Korak 1: vrh 1 dobiva oznaku  $TO$  i  $D_1 = 0$ . Vrh  $j (= 2, 3, \dots, n)$  dobiva oznaku  $PO$  i  $\tilde{D}_j = d(1, j) (= \infty$  ako brid  $(1, j) \notin E)$ . Stavlja se  $S_{TO} = \{1\}$ ,  $S_{PO} = \{2, 3, \dots, n\}$ .

Korak 2: pronalazi se  $k$  u  $S_{PO}$  za koji je  $\tilde{D}_k$  najmanja. Stavlja se  $D_k = \tilde{D}_k$ . Uzima se najmanji  $k$  ako ih ima više. Briše se  $k$  iz  $S_{PO}$  i uključuje u  $S_{TO}$ . Ako je  $S_{PO} = \emptyset$  tada

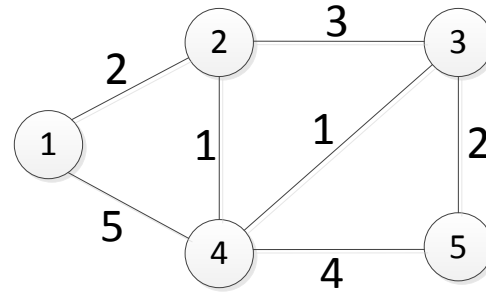
IZLAZ  $D_2, \dots, D_n$ . STOP. Inače

Korak 3: Za svaki  $j$  u  $S_{PO}$  stavlja se  $\tilde{D}_j = \min_k \{ \tilde{D}_j, D_k + d(k, j) \}$ . Vratiti se na korak 2.



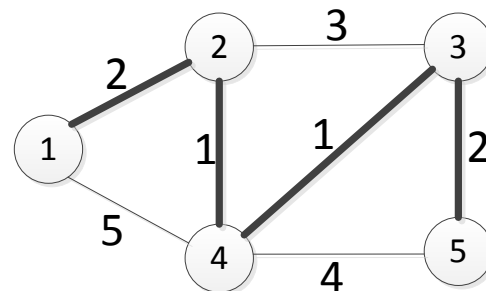
Slika 4. Dijagram toka za Dijkstra algoritam

Izračunavanje udaljenosti Dijkstra algoritmom od vrha 1 do ostalih vrhova grafa  $G$ , kao na slici 5., prezentirano je u nastavku.



Slika 5. Graf  $G$

Početno stanje je  $D_1 = 0$ ,  $\tilde{D}_2 = 2$ ,  $\tilde{D}_3 = \infty$ ,  $\tilde{D}_4 = 5$ ,  $\tilde{D}_5 = \infty$ . Stavlja se  $S_{TO} = \{1\}$ ,  $S_{PO} = \{2, 3, 4, 5\}$ . Za  $k = 2$  stavlja se  $D_2 = \tilde{D}_2 = 2$ .  $S_{TO} = \{1, 2\}$ ,  $S_{PO} = \{3, 4, 5\}$ . Zatim se računa  $\tilde{D}_3 = \min_2 \{ \tilde{D}_3, D_2 + d(2, 3) \} = \min_2 \{ \infty, 2 + 3 \} = 5$ ,  $\tilde{D}_4 = \min_2 \{ \tilde{D}_4, D_2 + d(2, 4) \} = \min_2 \{ 5, 2 + 1 \} = 3$ ,  $\tilde{D}_5 = \min_2 \{ \tilde{D}_5, D_2 + d(2, 5) \} = \min_2 \{ \infty, 2 + \infty \} = \infty$ . Za  $k = 4$  stavlja se  $D_4 = \tilde{D}_4 = 3$ .  $S_{TO} = \{1, 2, 4\}$ ,  $S_{PO} = \{3, 5\}$ . Računa se  $\tilde{D}_3 = \min_4 \{ \tilde{D}_3, D_4 + d(4, 3) \} = \min_4 \{ 5, 3 + 1 \} = 4$ ,  $\tilde{D}_5 = \min_4 \{ \tilde{D}_5, D_4 + d(4, 5) \} = \min_4 \{ \infty, 3 + 4 \} = 7$ . Za  $k = 3$  stavlja se  $D_3 = \tilde{D}_3 = 4$ .  $S_{TO} = \{1, 2, 4, 3\}$ ,  $S_{PO} = \{5\}$ . Računa se  $\tilde{D}_5 = \min_3 \{ \tilde{D}_5, D_3 + d(3, 5) \} = \min_3 \{ 7, 4 + 2 \} = 6$ . Za  $k = 5$  stavlja se  $D_5 = \tilde{D}_5 = 6$ .  $S_{TO} = \{1, 2, 4, 3, 5\}$ ,  $S_{PO} = \emptyset$ . Konačno,  $D_2 = 2$ ,  $D_3 = 4$ ,  $D_4 = 3$ ,  $D_5 = 6$ . Najkraći putovi od vrha 1 do svih vrhova težinskog grafa  $G$  nacrtani su debelom linijom na slici 6.



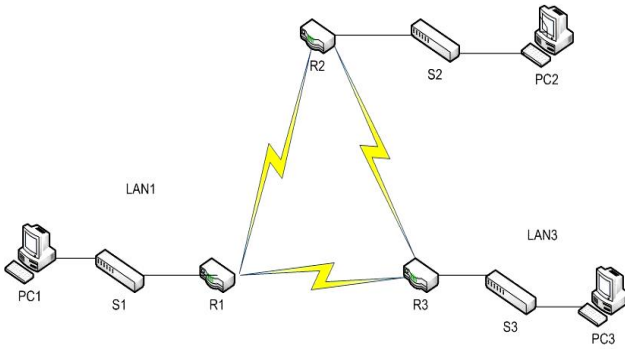
Slika 6. Najkraći putovi

Algoritam je vrlo sličan Bellman-Fordovom algoritmu. Razlika je u tome što Dijkstrin algoritam ne može raditi s negativnim vrijednostima. Za Bellman-Ford algoritam, algoritam provjerava vrijednosti točaka. Ako postoji negativna vrijednost, algoritmu se javlja greška. Dijkstrin algoritam pronalazi najkraće putove od početne točke do svih točaka u mreži [9].

## 7. KONFIGURACIJA

Fizička realizacija mreže je izvedena korištenjem CISCO-vih usmjernika i korištenjem aplikativnog programa PuTTY, kojim su upisivane potrebne naredbe namjenjene CISCO-vim operativnim sustavima. PuTTY je besplatni Telnet/SSH korisnički program, a riječ je o programu koji preko serijske komunikacije računala i usmjernika omogućuje parametriranje i programiranje usmjernika i preklopnika (engl. switch).

Slika 7. prikazuje fizičku topologiju izgrađene mreže za potrebe parametriranja i usporedbe EIGRP i OSPF protokola.



Slika 7. Topologija korištena u konfiguraciji protokola [4]

U prvoj fazi je potrebno izvršiti osnovnu konfiguraciju na svakom usmjerniku (konfiguracija imena, lozinka, banner). Zatim je potrebno konfigurirati sučelja svakog usmjernika te sam protokol usmjeravanja. Poželjno je korištenje naredbi koje omogućuju provjeru susjedstva i ruta uključenih u ažuriranja.

```
COM1 - PuTTY
R1(config)#
R1(config)#
R1(config)#
R1(config)#interface fa0/0
R1(config-if)#ip address 172.16.2.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
*Mar 22 09:26:04.803: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 22 09:26:05.803: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#interface s0/1/0
R1(config-if)#ip address 192.168.1.18 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
*Mar 22 09:26:42.183: %LINK-3-UPDOWN: Interface Serial0/1/0, changed state to down
R1(config-if)#interface s0/1/1
R1(config-if)#ip address 192.168.1.25 255.255.255.252
R1(config-if)#clock rate 64000
R1(config-if)#no shutdown
R1(config-if)#
*Mar 22 09:27:41.003: %LINK-3-UPDOWN: Interface Serial0/1/1, changed state to down
```

Slika 8. Primjeri konfiguracije serijskih i lokalnih sučelja

```
COM1 - PuTTY
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#router ospf 1
R1(config-router)#network 172.16.2.0 0.0.0.255 area 0
R1(config-router)#network 192.168.1.16 0.0.0.3 area 0
R1(config-router)#network 192.168.1.24 0.0.0.3 area 0
R1(config-router)#end
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
*Mar 28 09:17:04.535: %SYS-5-CONFIG I: Configured from console by console
```

Slika 9. Primjer konfiguracije OSPF protokola i mreža uključenih u OSPF ažuriranja

Da bi se neka ruta smatrala dostupnom, dostupnost treba provjeriti naredbom *ping*.

## 8. ZAKLJUČAK

Upotreba sve većih i složenijih dinamičkih računalnih mreža, međusobno umrežavanje različitih uređaja, povezivanje različitih mreža u složene strukture te rastuća ovisnost čovječanstva o mrežnim tehnologijama naglašavaju važnost protokola usmjeravanja. Od velike je važnosti brzo i točno dostaviti poslana podatke kako bi se povećala efikasnost mreže.

S obzirom na različite topologije, potrebe i prirode procesa, treba pažljivo odabrati odgovarajući protokol usmjeravanja. Zbog kompleksnosti problema usmjeravanja treba se prilikom modeliranja i implementacije protokola usmjeravanja držati općih smjernica unutar OSI modela.

Statičko usmjeravanje je pogodno za primjenu u manjoj lokalnoj mreži, ali nikako ne može izvršavati složene zadatke usmjeravanja unutar neke veće računalne mreže. Distance Vector protokoli pronalaze moguće najbolje putove na osnovu dostupnih informacija o udaljenosti do odredišta. Susjedima se šalje cijela tablica usmjeravanja. Ovi protokoli su jednostavni i učinkoviti u malim mrežama i vrlo su lagani za parametriranje i održavanje. U velikim i srednje velikim mrežama imaju loša konvergenzijska svojstva pa su zato razvijeni Link State protokoli s boljim performansama.

Kod Link State protokola usmjernik poznaje topologiju mreže i ne šalje cijelu tablicu usmjeravanja, nego se svim usmjernicima u mreži šalje samo informacija o stanju veze u obliku malih LSA paketa. Na temelju tih informacija usmjerivači mogu kvalitetno izračunati sve raspoložive rute. Ova metoda je pouzdanija, troši manje pojase širine mreže, odnosno propusnosti, a konvergencija je brža.

Negativna strana Link State protokola je kompleksniji algoritam, što znači i veće opterećenje procesora usmjernika i veća potrošnja memorije, naročito prilikom početne konvergenije kada usmjernici imaju prazne tablice usmjeravanja.

Protokoli usmjeravanja su vrlo kompleksna tema o kojoj se može mnogo pisati i detaljno je analizirati sukladno konfiguracijama zatečenim u praksi. Radi se o veoma zanimljivom području koje će se zasigurno u budućnosti dalje razvijati i unapređivati. S vremenom će se sigurno pojaviti mnoštvo novih i boljih protokola, algoritama, novih načina razmišljanja i pristupanja problemima s kojima će biti susretanja u konačnom svijetu beskonačnih mogućnosti umrežavanja.

## 9. LITERATURA

- [1] Knez, K.: Usporedba i primjena „Distance Vector“ i „Link State“ mrežnih protokola, Završni rad br. 275/EL/2012, VELV Varaždin, 2013.
- [2] Kurose, J. F.; Ross, K. W.: Computer Networking, Pearson Education, 2012.
- [3] [http://hr.wikipedia.org/wiki/OSI\\_model](http://hr.wikipedia.org/wiki/OSI_model) (Dostupno: 04.03.2013.)
- [4] <http://www.cisco.com/web/learning/netacad/index.html> (Dostupno:04.03.2013.)
- [5] Odom W.: Cisco CCNA Routing and Switching, Cisco press, 2010.
- [6] Browning P., Subnetting Secrets, Reality press, 2012.
- [7] Cavendish D.; Gerla M.: Internet QoS Routing using the Bellman-Ford Algorithm, Chapman & Hall, 1998.
- [8] Graziani R.; Johnson A.: Routing Protocols and Concepts, Cisco press, 2008.
- [9] [http://en.wikibooks.org/wiki/Artificial\\_Intelligence/Search/Dijkstra's\\_Algorithm](http://en.wikibooks.org/wiki/Artificial_Intelligence/Search/Dijkstra's_Algorithm) (Dostupno:10.04. 2013.)
- [10] Veljan D.: Kombinatorika s teorijom grafova, Školska knjiga, Zagreb, 1989.
- [11] Kreyszig E.: Advanced engineering mathematics – 10th ed., John Willey & Sons, 2010.

### Kontakt autora:

**mr. sc. Ladislav Havaš**  
Veleučilište u Varaždinu  
J. Križanića 33, 42 000 Varaždin  
[ladislav.havas@velv.hr](mailto:ladislav.havas@velv.hr)

**Damira Keček, dipl. ing.**  
Veleučilište u Varaždinu  
J. Križanića 33, 42 000 Varaždin  
[damira.kecek@velv.hr](mailto:damira.kecek@velv.hr)

**Kristijan Knez (bivši student)**  
Veleučilište u Varaždinu  
J. Križanića 33, 42 000 Varaždin  
[krknez@velv.hr](mailto:krknez@velv.hr)