ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU

IMPLEMENTACIJA I ANALIZA ALGORITAMA ZA KREIRANJE STABLA SA PRIORITETIMA

–Master rad–

Kandidat:

Mihailo Čelebić 2011/3413

Mentor:

doc. dr Zoran Čiča

Beograd, Septembar 2014.

SADRŽAJ

1. UVOD

Ruteri predstavljaju najbitnije elemente infrastrukture Interneta. Osnovna funkcija rutera je prosleđivanje IP paketa između mreža. Prosleđivanje paketa na odgovarajući port rutera vrši se na osnovu odredišne IP adrese pristiglog paketa i lukap tabele. Određivanje izlaznog porta rutera na koji treba proslediti pristigli IP paket se naziva lukap funkcija. Ogromno povećanje lukap tabela i brzine pristizanja paketa zahteva veoma brzo izvršavanje lukap funkcije. Prelaz na duže IPv6 adrese takođe povećava lukap tabele. Otuda lukap funkcija mora da omogući efikasnu i brzu pretragu, pri čemu lukap tabela mora da bude formatirana tako da ne zauzima veliki prostor kako bi se omogućilo korišćenje bržih memorija za smeštanje lukap tabele.

IP lukap funkcija može koristiti binarno stablo kao vrlo jednostavnu strukturu podataka. Kod ovakvih algoritama lukap tabela predstavljena je u obliku binarnog stabla. Na kraju svake putanje u stablu određene mrežnim prefiksom iz lukap tabele nalazi se „popunjen čvor“ koji sadrži informaciju o izlaznom portu na koji treba usmeriti pakete sa odredišnom IP adresom za koju taj prefiks predstavlja prefiks sa najdužim poklapanjem u lukap tabeli. Stablo sa prioritetima predstavlja unapređenje klasičnog binarnog stabla kao bazične strukture za predstavu lukap tabele. Binarno stablo u slučaju velikih lukap tabela sadrži ogroman broj praznih čvorova koji ne sadrže korisne informacije (mrežne prefikse) čime se značajno povećavaju memorijski zahtevi, naročito u slučaju IPv6 tabela. Stablo sa prioritetima transformiše klasično binarno stablo u binarno stablo koje sadrži samo popunjene čvorove (tj. stablo sa prioritetima). Transformacija se postiže pomeranjem listova stabla u prazne čvorove, do eliminacije svih praznih čvorova u stablu. List čvorom nazivamo čvor koji nema „potomaka“ (decu). Termin sa prioritetima označava da se prilikom pretrage lukap tabele u slučaju nalaska rešenja pretraga može okončati jer se sigurno ne može naći bolje rešenje, za razliku od klasičnih binarnih stabala gde se pretraga nastavlja zbog mogućnosti nalaženja boljeg rešenja. Transformacijom binarnog stabla u stablo sa prioritetima oslobođa se značajan deo memorijskih resursa, smanjuje se dubina stabla i vrši preraspodela čvorova što rezultuje povećanjem performansi lukap funkcije. Cilj ovog rada je implementacija i analiza performansi različitih algoritama formiranja stabla sa prioritetima.

U drugom poglavlju ovog rada biće reči o binarnom stablu i stablu sa prioritetima kao strukuturama podataka koje lukap funkcija može da koristi. U trećem poglavlju biće predloženo nekoliko tehnika formiranja stabla sa prioritetima. Takođe, u trećem poglavlju biće predstavljena i implementacija predloženih tehnika formiranja stabla sa prioritetima. Tehnike formiranja stabla biće softverski realizovane u programskom jeziku C. U četvrtom poglavlju biće data analiza performansi stabala sa prioritetima kreiranih pomoću tehnika opisanih u trećem poglavlju.

2. STABLA SA PRIORITETIMA

IP lukap funkcija je funkcija rutera koja na osnovu odredišne IP adrese dolaznog IP paketa vrši izbor izlaznog porta rutera na koji prosleđuje taj paket kako bi stigao do krajnje destinacije. Izbor izlaznog porta rutera vrši se na osnovu definisane tabele rutiranja (*RIB - routing information base*). Tabela rutiranja sadrži najmanje tri vrste podataka: IP prefiks, metrika (cena) i sledeći hop (*gateway* adresa, izlazni port rutera). Kod svake IP adrese razlikujemo dva dela: mrežni deo i host deo. Klasno adresiranje podrazumeva da mrežni deo IP adrese može imati tri fiksne dužine (8, 16 i 24 bita). Zbog neracionalnog korišćenja adresnog prostora, na Internetu je klasno adresiranje ubrzo zamenjeno besklasnim adresiranjem. Kod besklasnog adresiranja granica između mrežnog dela host dela je proizvoljna pa je i korišćenje adresnog prostora efikasnije. Takođe, besklasno adresiranje omogućuje agregiranje više mrežnih adresa u jedan zapis u tabeli rutiranja. Pošto se u tabeli rutiranja mogu nalaziti mrežne adrese, ali i agregacije mrežnih adresa, usvojen je termin prefiks koji označava oba slučaja i u nastavku rada će se koristiti ovaj termin. Primer jedne IP adrese i njoj odgovarajućeg prefiksa dat je u tabeli 2.1.

Tabela 2.1 Primer IP adrese i odgovarajućeg prefiksa

IP adresa	109.93.44.210	**01101101010111010010110**0|11010010
Mrežna maska	255.255.255.0	**11111111111111111111111**1|00000000
Prefiks	109.93.44.0	**01101101010111010010110**0|00000000

Agregacija mrežnih adresa značajno smanjuje broj zapisa u lukap tabelama i samim tim njihovu veličinu, ali dovodi do pojave da za jednu IP adresu može da postoji više rešenja. Ova pojava značajno doprinosi kompleksnosti lukap funkcije jer prilikom nalaženja rešenja neophodno je biti siguran i da je ono najbolje rešenje jer u suprotnom može doći do grešaka u usmeravanju. Za rešenje ovog problema usvojeno je pravilo najdužeg poklapanja koje određuje da se za konačno rešenje usvoji zapis koji se najduže poklapa sa IP adresom za koju se vrši lukap funkcija. Ovo pravilo se naziva LPM (*Longest Matching Prefix*) pravilo. [1]

U mreži se neprestano dešavaju promene u topologiji. Promene mogu nastati usled dodavanja novih linkova, rutera i dr., ali i usled otkaza istih. Da bi IP paketi nastavili da stižu na ispravne destinacije, protokoli rutiranja moraju obavestiti sve rutere u mreži o nastalim izmenama. Zatim, ruteri vrše ažuriranje tabela rutiranja u okviru koga se dodaju novi prefiksi, brišu postojeći prefiksi ali i menja informacija o izlaznom portu za postojeće prefikse. Da bi što ređe dolazilo do grešaka u usmeravanju podataka usled netačnosti podataka u lukap tabeli neophodno je da se proces ažuriranja lukap tabele izvršava najvećom mogućom brzinom.

Veliki problem u realizaciji lukap funkcije predstavlja veliki broj prefiksa. Tabele rutiranja mogu imati i preko 500000 prefiksa. Samim tim, kreiranje efikasne lukap funkcije predstavlja veliki izazov. Takođe, neminovan je i prelaz na IPv6 adrese, što dodatno povećava prostor za pretragu. Pored povećanja tabela rutiranja, dramatično raste i Internet saobraćaj, što dovodi do povećanja brzine linkova a samim tim i brzine pristizanja paketa na ruterima. Zbog toga se od rutera zahteva

da izvrši lukap funkciju za što kraće vreme – u vremenu od nekoliko nanosekundi u slučaju linkova brzine 100Gb/s.

Performanse IP lukap funkcije zavise od nekoliko faktora. Primarni faktor je brzina pretraživanja lukap tabele koja najviše zavisi od broja pristupa memoriji, obzirom da je pristup memoriji najsporija operacija tokom pretrage. Kako tabele rutiranja dramatično rastu, bitan faktor je veličina potrebne memorije. Takođe, veoma je bitna i sposobnost brze izmene tabele rutiranja kako bi se izbeglo slanje IP paketa na pogrešnu destinaciju. Skalabilnost je takođe bitan faktor obzirom na stalni rast tabela rutiranja.

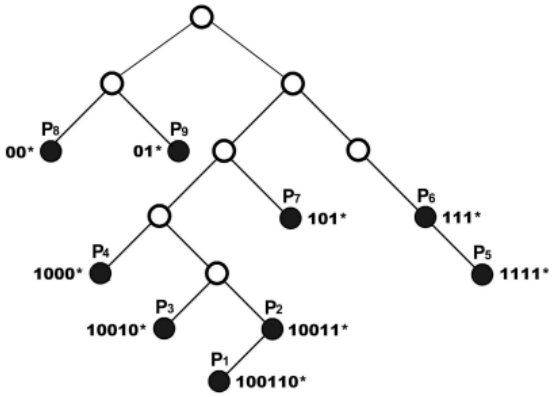
Lukap algoritam definiše način pretrage lukap tabele, ali i definiše strukturu lukap tabele. Razlikujemo tri glavne klase lukap algoritama [1]:

- Lukap algoritmi zasnovani na strukturi stabla
- Lukap algoritmi zasnovani na TCAM (*Ternary Content-Addressable Memory*) memorijama
- Lukap algoritmi bazirani na heširanju

U ovom radu, biće predstavljeni algoritmi isključivo zasnovani na strukturi stabla jer u tu klasu spadaju stabla sa prioritetima.

2.1. Binarna stabla

IP lukap funkcija može koristiti binarno stablo kao veoma jednostavnu strukturu podataka. Kod ovakvih algoritama tabela rutiranja predstavljena je u obliku binarnog stabla. IP prefiksi se mogu predstaviti u binarnom obliku dužine do 32 bita za slučaj IPv4 adresa (napomena: 128b za IPv6 slučaj). Svaki prefiks predstavlja putanju do čvora, pri čemu dubina čvora odgovara dužini prefiksa. Na kraju svake putanje određene prefiksom nalazi se „popunjen čvor“ koji sadrži informaciju o izlaznom portu na koji treba usmeriti pakete sa odredišnom IP adresom za koju taj prefiks predstavlja prefiks sa najdužim poklapanjem u lukap tabeli. Ovako kreirano stablo može imati dubinu do 32. Prilikom pretrage vrši se kretanje kroz stablo na osnovu IP adrese tako što vrednost uzetog bita IP adrese ‘0’ ukazuje na smer levo, dok vrednost bita ‘1’ ukazuje na smer desno. Prvi upit se vrši na korenu stabla na osnovu bita najveće težine (1. bit). Svaki čvor binarnog stabla ima fiksnu poziciju definisanu binarnim nizom (tj. IP prefiksom). Čvor koji nema „potomaka“ (decu) nazivamo list čvorom. Na slici 2.1.1 prikazan je primer binarnog stabla na osnovu malog seta IP prefiksa.



Slika 2.1.1. Primer binarnog stabla[2]

Binarno stablo predstavlja veoma jednostavno rešenje za implementaciju lukap funkcije. Međutim, u slučaju velikih lukap tabela binarno stablo sadrži ogroman broj praznih čvorova koji ne sadrže korisne informacije (IP prefikse). Ovo je naročito izraženo kod IPv6 tabela obzirom da je dužina IPv6 adrese 128 bita. Ogroman broj čvorova značajno povećava memorijske zahteve i smanjuje brzinu pretraživanja. Broj praznih čvorova zavisi isključivo od seta prefiksa u lukap tabeli. Takođe, kod binarnih stabala kraći prefiksi se nalaze na nižim nivoima od dužih prefiksa. Zbog toga, trajanje pretrage za kraće prefikse je brže nego za duže prefikse. U ovom dokumentu nižim nivoima smatramo nivo bliže korenu stabla. Jedna ulazna IP adresa može imati više prefiksa sa kojima se poklapa. Prema tome, iako je nađen prefiks koji se poklapa sa ulaznom adresom, pretraga se mora nastaviti do list čvora jer postoji mogućnost da postoji duži prefiks koji odgovara ulaznoj adresi. Ovo značajno umanjuje efikasnost pretraživanja.

Ažuriranje strukture binarnog stabla je veoma jednostavno. Proces dodavanja novog prefiksa se svodi na upis informacije o odgovarajućem izlaznom portu u postojeći čvor, ili na kreiranje novih čvorova. Brisanje čvorova se svodi na uklanjanje dotičnog čvora iz stabla i eventualno uklanjanje praznih čvorova koji nemaju druge potomke.

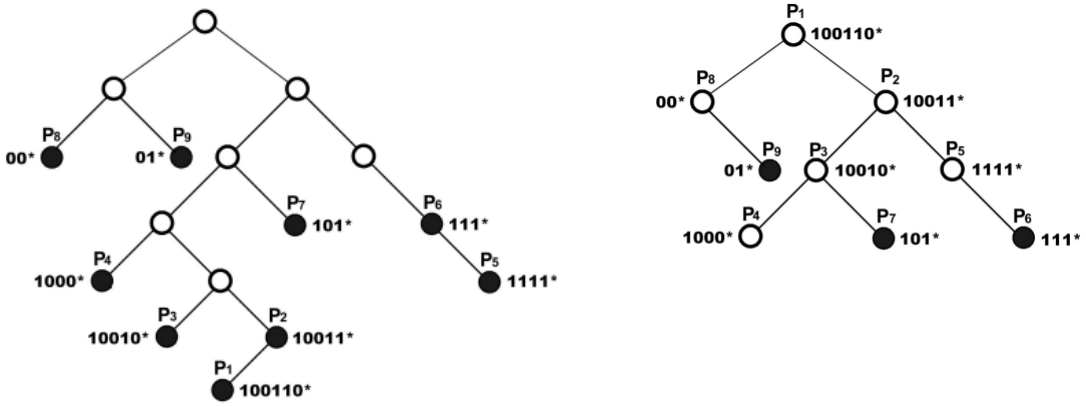
2.2. Stabla sa prioritetima

Stabla sa prioritetima za potrebe lukap funkcije predložena su u radu "Priority Tries for IP Address Lookup"[2]. Stablo sa prioritetima predstavlja unapređenje klasičnog binarnog stabla kao bazične strukture za predstavu lukap tabele. Unapređenje se pre svega odnosi na eliminaciju praznih čvorova koji ne nose korisnu informaciju, a koriste memorijske resurse. Pored problema sa velikim brojem praznih čvorova, još jedan problem je to što su duži prefiksi smešteni dublje u stablo, pa se upoređuju kasnije sa ulaznom IP adresom. Pretraga se nastavlja do list čvora, iako je nađen prefiks koji se poklapa sa ulaznom IP adresom. Kada bi prefiksi bili obrnuto smešteni, odnosno kada bi duži prefiksi bili smešteni u nižim nivoima a kraći prefiksi u višim nivoima, pretraga bi se završavala odmah po prvom poklapanju sa ulaznom IP adresom.

Čvorovi stabla sa prioritetima pored informacije o vrednosti izlaznog porta i dva pokazivača moraju da sadrže i dodatne podatke. Jedan podatak je binarni indikator da li u pitanju prioritetni (pomereni) čvor ili ne. Drugi podatak je vrednost samog prefiksa.[1]

Stablo sa prioritetima se kreira tako što se list čvorovi smeštaju u prazne čvorove najmanje dubine. Pri tome, prazan čvor u koji se smešta list čvor mora biti deo putanje do list čvora, odnosno mora biti „predak" list čvora. Prazan čvor u koji se smešta list čvor (odnosno prefiks list čvora)

nazivamo prioritetnim čvorom. List čvor koji se premešta, se nakon premeštanja uklanja iz stabla sa svoje inicijalne pozicije. Takođe, ukoliko list čvor ima prazne pretke koji nemaju drugih potomaka, oni se takođe uklanjaju iz stabla. Postupak se ponavlja do eliminisanja svih praznih čvorova u stablu. Primer ovakve transformacije binarnog stabla u stablo sa prioritetima dat je na slici 2.2.1.



Slika 2.2.1 Transformacija binarnog stabla (levo) u stablo sa prioritetima (desno)[2]

Za razliku od binarnog stabla, stablo sa prioritetima nema prazne čvorove koji nepotrebno zauzimaju memorijske resurse. Ovime se značajno štedi memorijski prostor, naročito kada su u pitanju IPv6 tabele. Smanjivanjem broja čvorova može doći i do smanjenja dubine stabla, pa se smanjuje i maksimalan broj poređenja ulazne IP adrese sa prefiksom u najgorem slučaju. Takođe, poređenje ulazne IP adrese sa dužim prefiksima se vrši ranije, pa se prilikom prvog poklapanja pretraga može okončati. Ovime značajno rastu performanse lukap funkcije jer se smanjuje broj pristupa memoriji.

Kod stabla sa prioritetima određivanje izlaznog porta IP paketa na osnovu odredišne IP adrese vrši se na sledeći način. I dalje se vrši prolazak kroz stablo na osnovu IP adrese po principu „bit po bit“. Međutim, kada se dođe do prioritetnog čvora (čvor u koji je smešten prefiks list čvora) vrši se provera da li taj prefiks odgovara odredišnoj IP adresi. Ukoliko ne odgovara, u zavisnosti od vrednosti bita kreće se ka levom ili desnom čvoru detetu kao kod klasičnog binarnog stabla. Ukoliko odgovara, pretraga se okončava jer vrednost unetog prefiksa sigurno predstavlja najduže poklapanje sa odredišnom IP adresom. Prema tome, može se reći da duži prefiksi imaju veći prioritet za razliku od klasičnog binarnog stabla, pa ovakvo stablo nazivamo stablom sa prioritetima.

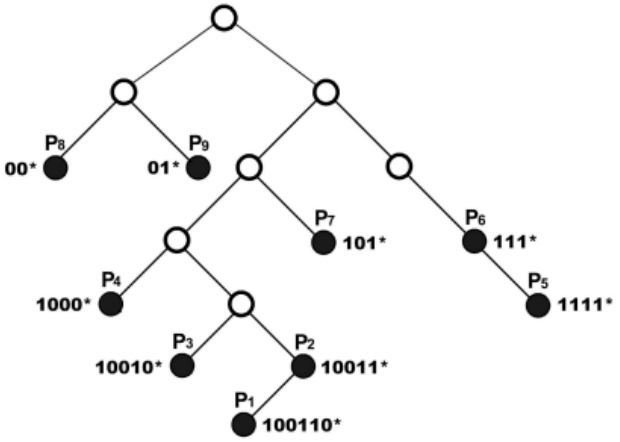
I dalje ostaje problem višestrukog pristupa memoriji iz klasičnih binarnih stabala jer se u najgorem slučaju mora ići do lista stabla sa prioritetima tokom pretrage. U slučaju da se želi primeniti pajplajn tehnika gde bi svaki nivo stabla išao u zasebnu memoriju, problem predstavlja nejednaka distribucija čvorova po nivoima što otežava efikasnu hardversku implementaciju pajplajn tehnike jer se ne može koristiti isti tip memorije za sve pajplajn faze (nivoe stabla) u implementaciji. Ovi problemi se mogu prevazići kombinovanjem stabla sa prioritetima sa drugim tehnikama.

3. IMPLEMENTACIJA TEHNIKA ZA FORMIRANJE STABLA SA PRIORITETIMA

U ovom delu rada biće reči o tehnikama za formiranje stabla sa prioritetima, tačnije o tehnikama transformacije binarnog stabla u stablo sa prioritetima. Najpre ćemo definisati kriterijume koji će se koristiti u izboru list čvora koji se premešta. Zatim ćemo opisati različite tehnike formiranja stabla sa prioritetima. Transformacija binarnog stabla u stablo sa prioritetima biće softverski realizovana u programskom jeziku C. U poslednjem delu ovog poglavlja daćemo kratak opis kodova implementacije prethodno opisanih tehnika.

Kriterijumi za izbor čvora koji se premešta mogu biti različiti. Ključno je da čvor koji se premešta mora biti list čvor koji ima barem jedan prazan čvor u putanji do njega (barem jednog praznog pretka). List čvor se sa svoje pozicije premešta u praznog pretka najmanje dubine. Nakon toga, ukoliko čvor koji premeštamo (list čvor) ima prazne pretke koji nemaju druge potomke osim njega, i oni se uklanjaju iz stabla zajedno sa njim. Prilikom uklanjanja praznih predaka koji nemaju potomaka (osim list čvora koji izmeštamo), treba voditi računa o uslovu da mora postojati barem jedan prazan čvor (predak) u koji premeštamo list čvor. U nastavku ćemo opisati kriterijume za izbor list čvora, koji su korišćeni u ovom radu.

Dubina čvora je jedan od korišćenih kriterijuma i može se predstaviti kao broj hopova od korena stabla do čvora koji posmatramo. Na slici 3.1 imamo primer jednog binarnog stabla. Ovo stablo ima 2 čvora dubine 1, 4 čvora dubine 2, 3 čvora dubine 3, 3 čvora dubine 4, 2 čvora dubine 5 i jedan čvor dubine 6. Dubina stabla odgovara najvećoj dubini čvora tog stabla. Za stablu sa slike 3.1 dubina stabla je 6.



Slika 3.1. Primer binarnog stabla[2]

Sa aspekta korena stabla sa slike, dubina levog podstabla je 2, dok je dubina desnog podstabla 6. Ukoliko se koristi kriterijum veće dubine, na svakom čvoru vrši se upit da li je veća

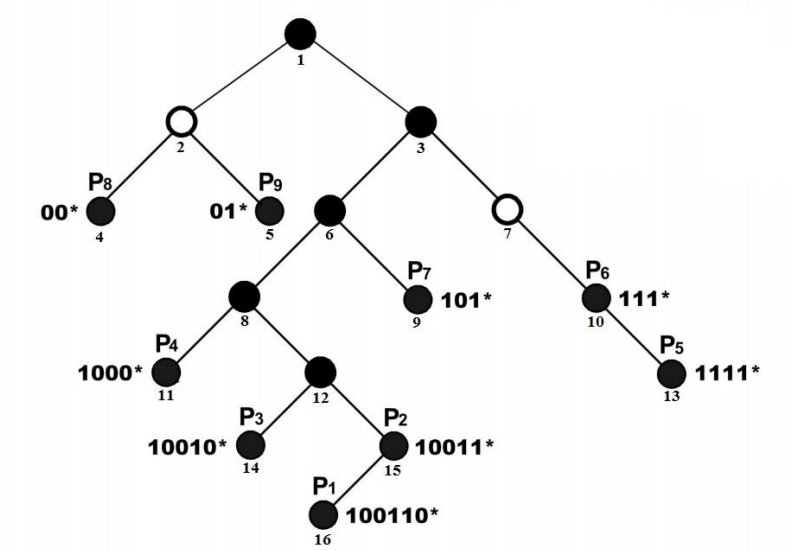
dubina levog ili desnog podstabla, i vrši se kretanje ka levo ako je dubina levog podstabla veća odnosno ka desno ako je dubina desnog podstabla veća.

U stablu na slici 3.1., broj čvorova je 16, broj praznih čvorova je 7 dok je broj popunjenih čvorova 9. U ovom radu za izbor list čvora korišćeni su i kriterijumi zasnovani na broju čvorova i na broju praznih čvorova. Sa aspekta korena stabla, u levom podstablu nalazi se 3 čvora od kojih je 1 prazan čvor, dok se u desnom podstablu nalazi 12 čvorova od kojih je 5 čvorova prazno. Ukoliko se kao kriterijum koristi veći broj čvorova, na svakom čvoru se vrši upit da li je veći broj čvorova veći u levom ili u desnom podstablu i vrši kretanje ka levo ako je u levom podstablu veći broj čvorova, odnosno ka desno ako je u desnom podstablu veći broj čvorova. Analogno, kretanje kroz stablo od korena do list čvora se može vršiti na osnovu broja praznih čvorova.

Broj praznih predaka je takođe kriterijum za izbor list čvora koji je korišćen u ovom radu. Svaki čvor u stablu mora imati ažurnu vrednost broja praznih predaka. Na slici 3.1., list čvor najveće dubine ima 5 praznih predaka. Za razliku od prethodno definisanih kriterijuma, izbor list čvora nije moguće vršiti kretanjem kroz stablo počev od korena stabla. Za izbor svakog list čvora koji izmeštamo prema kriterijumu broja praznih predaka, potrebno je ažurirati vrednost praznih predaka za svaki čvor u stablu. Nakon provere svih čvorova, za list čvor uzima se čvor sa najvećom ili najmanjom vrednošću broja praznih predaka zavisno od kriterijuma.

Kretanje kroz stablo u cilju izbora list čvora koji se izmešta moguće je i metodom slučajnog izbora. Na svakom čvoru počev od korena stabla vrši se kretanje ka levo ili ka desno po principu slučajnog izbora.

Bez obzira na kriterijume koji se koriste, kako bi se transformacija binarnog stabla u stablo sa prioritetima uspešno izvršila, neophodno je da svaki list čvor koji izmeštamo ima barem jednog praznog pretka u koji će taj čvor biti smešten. Ovo se može smatrati najprioritetnijim kriterijumom kod svih tehnika formiranja stabla sa prioritetima. Razlog za to ćemo pokazati na primeru stabla datom na slici 3.2. Smatraćemo da je dubina osnovni kriterijum za izbor list čvora koji se premešta. Na korenu stabla (čvor br.1) pravimo upit, da li je dubina veća u levom ili u desnom podstablu. Prema tome, krećemo se ka desno, odnosno dolazimo na čvor br.3. Na čvoru br.3, ponovo vršimo upit gde je dubina stabla veća, i krećemo se ka levo. Ponavljanjem ovog postupka dolazimo do list čvora br.16. Međutim, čvor br.16 nema prazne pretke, samim tim, nemamo gde da premestimo ovaj čvor. Prema tome, ukoliko prilikom prolaska kroz stablo nismo prošli preko praznog čvora, pre osnovnog kriterijuma potrebno je proveriti da li oba podstabla sadrže prazne čvorove. Ukoliko sadrže, možemo koristiti osnovni kriterijum. Ukoliko nemamo praznih čvorova u oba podstabla, krećemo se u smeru u kom ima praznih čvorova. U primeru na slici 3.2., na čvoru br.3, iako osnovni kriterijum upućuje na kretanje u levo, potrebno je izvršiti kretanje u desno jer tako obezbeđujemo prazan čvor za smeštanje prefiksa list čvora koji izmeštamo. Ukoliko smo u prolasku kroz stablo prešli preko jednog praznog čvora, nije potrebno voditi računa o broju praznih čvorova prilikom kretanja kroz ostatak stabla.



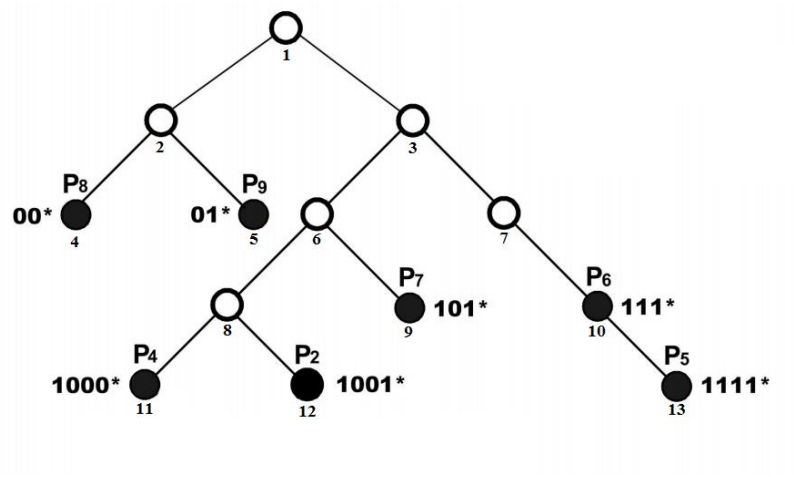
Slika 3.2. Primer stabla

U ovom radu, biće analizirano 11 tehnika za formiranje stabla sa prioritetima. Tehnike su nazvane prema kriterijumima izbora list čvora počev od kriterijuma najvećeg prioriteta. Tehnike koje će biti implementirane u okviru ovog rada su:

Tehnika 1.1. - Veća dubina, veći broj praznih čvorova, deterministički (levo)
Tehnika 1.2. - Veća dubina, veći broj praznih čvorova, statistički (slučajan izbor)
Tehnika 2.1. - Veća dubina, veći broj čvorova, slučajan izbor
Tehnika 2.2. - Veća dubina, manji broj čvorova, slučajan izbor
Tehnika 3.1. - Veći broj praznih čvorova, veća dubina, slučajan izbor
Tehnika 3.2. - Manji broj praznih čvorova, veća dubina, slučajan izbor
Tehnika 4.1. - Veći broj čvorova, veći broj praznih čvorova, slučajan izbor
Tehnika 4.2. - Manji broj čvorova, veći broj praznih čvorova, slučajan izbor
Tehnika 5. - Slučajan izbor
Tehnika 6.1. - Veći broj praznih predaka, veća dubina
Tehnika 6.2. - Manji broj praznih predaka, veća dubina

Tehnika 1.1. – Osnovni kriterijum izbora list čvora kod ove tehnike je veća dubina. Prilikom prolaska kroz stablo, na svakom čvoru vrši se upit da li je veća dubina levog ili desnog podstabla, i vrši se kretanje ka levo ako je dubina levog podstabla veća odnosno ka desno ako je dubina desnog podstabla veća. Ukoliko je dubina ista i u levom i u desnom podstablu, kriterijum koji se primenjuje je veći broj praznih čvorova (potomaka). Odnosno, ako je veći broj praznih potomaka u levom podstablu vrši se kretanje u levo, a ako je broj praznih potomaka veći u desnom podstablu vrši se kretanje ka desno. Ukoliko i dubina i broj praznih čvorova imaju iste vrednosti i levo i desno, kretanje se vrši u levo (deterministički određeno). Na slici 3.3. dat je karakterističan primer stabla za ovu tehniku formiranja stabla sa prioritetima. Na čvoru br.1 vrši se kretanje u desno (na čvor br.3) jer desno podstablo ima veću dubinu. Pošto je na čvoru br.3 dubina ista na obe strane, proverava se broj praznih čvorova u levom i u desnom podstablu. Kako je broj praznih čvorova veći u levom podstablu vrši se kretanje u levo. Zbog veće dubine, sa čvora br.6 vrši se kretanje ka čvoru br.8. Na čvoru br.8 proverom utvrđujemo da su i dubina i broj praznih čvorova isti u oba smera, što

je preduslov za treći kriterijum prolaska kroz stablo. Prema tom kriterijumu, vrši se kretanje u levo i dolazi do list čvora (čvor br.12) koji se premešta.

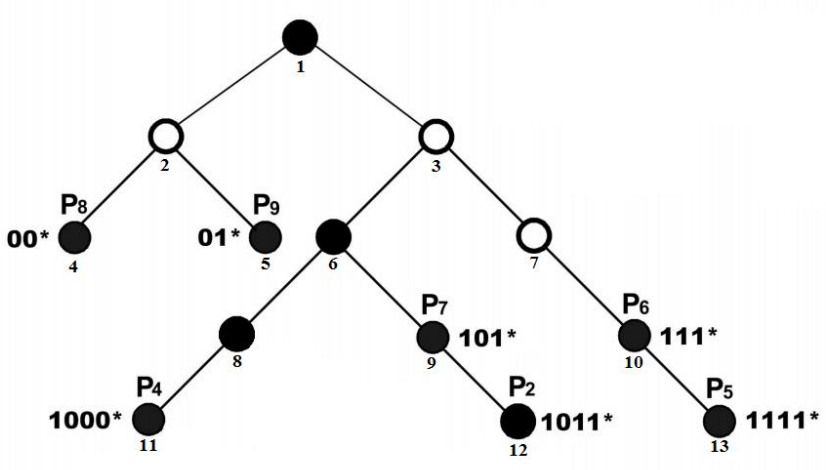


Slika 3.3. Karakterističan primer stabla za tehniku 1.1. i 1.2

Tehnika 1.2. – Ova tehnika za formiranje stabla sa prioritetima predstavlja malo izmenjenu verziju tehnike 1.1.. Kao kod tehnike 1.1. osnovni kriterijum za izbor list čvora je veća dubina, a kada je dubina ista u oba smera, vrši se kretanje u smeru gde ima više praznih čvorova. Jedina razlika u odnosu na tehniku 1.1. jeste treći kriterijum. Kada i dubina i broj praznih čvorova imaju iste vrednosti u oba smera, kretanje se vrši slučajnim izborom. Ako ponovo posmatramo primer dat na slici 3.3., razlika u odnosu na tehniku 1.1. uočljiva je na čvoru br.8. Na čvoru br.8 vrši se slučajan izbor. Prema tome, list čvor koji premeštamo biće ili čvor br.11 ili čvor br.12.

Analizom rezultata tehnike 1.1 i tehnike 1.2 u glavi 4, dolazi se do zaključka da je slučajan izbor kao poslednji kriterijum bolje rešenje od determinističkog. Zbog toga, u preostalim tehnikama (gde ima potrebe za tim) koristiće se slučajan izbor kao poslednji kriterijum.

Tehnika 2.1. – Osnovni kriterijum za izbor list čvora kod ove tehnike je takođe veća dubina. Kada je ista dubina levog podstabla i desnog podstabla, kretanje se vrši u smeru u kom ima više čvorova. Ukoliko je i dubina i broj čvorova isti i u levom i u desnom podstablu, odluka o kretanju desno ili levo donosi se metodom slučajnog izbora. Na slici 3.4. prikazan je karakterističan primer za ovu tehniku formiranja stabla sa prioritetima.

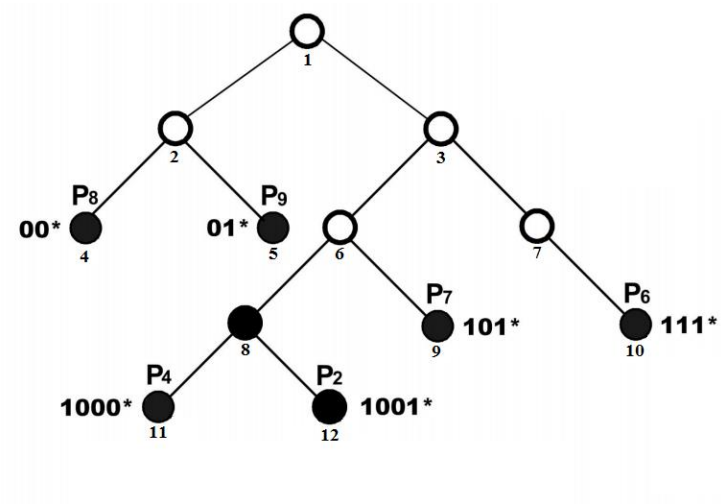


Slika 3.4 karakterističan primer stabla za tehnike 2.1 i 2.2

Na čvoru br.1 vrši se kretanje u desno (na čvor br.3) jer desno podstablo ima veću dubinu. Pošto je na čvoru br.3 dubina ista na obe strane, proverava se broj čvorova u levom i u desnom podstablu. Zbog većeg broja čvorova u levom podstablu, vrši se kretanje u levo ka čvoru br.6. Na čvoru br.6 možemo primetiti da je i dubina i broj čvorova u levom podstablu i u desnom podstablu jednak. U tom slučaju, kretanje vršimo metodom slučajnog izbora. Prema tome, list čvor koji premeštamo biće ili čvor br.11 ili čvor br.12..

Tehnika 2.2. – Ova tehnika za formiranje stabla sa prioritetima predstavlja malo izmenjenu verziju tehnike 2.1.. Kao kod tehnike 3.1. osnovni kriterijum za izbor list čvora je veća dubina, ali kada je dubina ista u oba podstabla, vrši se kretanje u smeru gde ima manje čvorova. Ovo predstavlja jedinu razliku u poređenju sa tehnikom 2.1.. Treći kriterijum je i dalje slučajan izbor. Ako ponovo posmatramo primer dat na slici 3.4, razlika u odnosu na tehniku 2.1 uočljiva je na čvoru br.3. Ovde se zbog iste dubine u levom i desnom podstablu vrši kretanje u desno zbog manjeg broja čvorova. Prema tome, list čvor koji se izmešta je u ovom slučaju čvor br.13.

Tehnika 3.1. – Osnovni kriterijum za izbor list čvora kod ove tehnike je veći broj praznih čvorova. Ukoliko je u levom i desnom podstablu broj praznih čvorova isti, kretanje u levo ili u desno vrši se na osnovu veće dubine. Ukoliko je i dubina ista u levom i u desnom podstablu, kretanje se vrši metodom slučajnog izbora. Na slici 3.5 prikazan je karakterističan primer za ovu tehniku formiranja stabla sa prioritetima. Na čvoru br.1 vrši se kretanje u desno (na čvor br.3) jer desno podstablo ima veći broj praznih čvorova. Na čvoru br.3 se vrši izbor po principu veće dubine jer je broj praznih čvorova isti. Prema tome, vrši se kretanje u levo ka čvoru br.6. Na čvoru br.6 kriterijum je takođe veća dubina obzirom da je broj praznih čvorova nula u oba smera. Na čvoru br.8 kretanje vršimo metodom slučajnog izbora obzirom da su brojevi praznih čvorova i dubine u oba smera iste. Prema tome, list čvor koji premeštamo biće ili čvor br.11 ili čvor br.12.

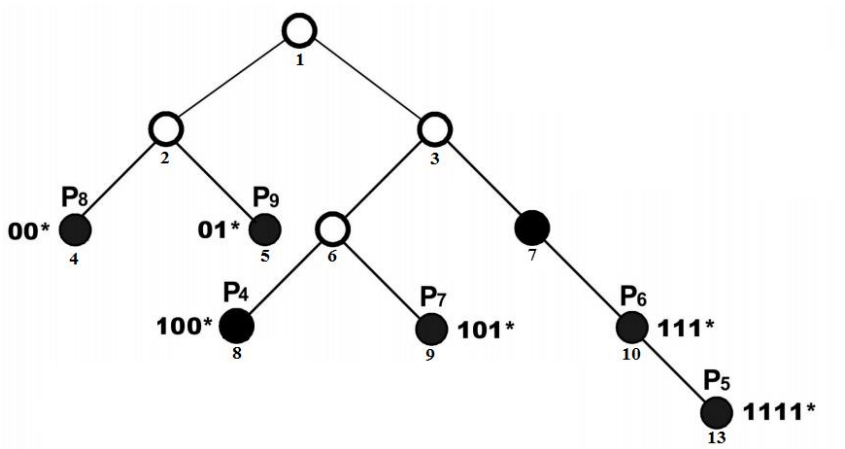


Slika 3.5 karakterističan primer stabla za tehnike 3.1, 3.2, 6.1 i 6.2

Tehnika 3.2. – Ova tehnika za formiranje stabla sa prioritetima predstavlja izmenjenu verziju tehnike 3.1.. Osnovni kriterijum za kretanje kroz stablo u cilju izbora list čvora je manji broj praznih čvorova. Ovo ujedno predstavlja i jedinu razliku u odnosu na tehniku 3.1.. Kao i kod tehnike 3.1., ukoliko je u levom i desnom podstablu broj praznih čvorova isti, kretanje u levo ili u desno vrši se na osnovu veće dubine. Ukoliko je i dubina ista u levom i u desnom podstablu, kretanje se vrši metodom slučajnog izbora. Ako ponovo posmatramo primer dat na slici 3.5, razlika u odnosu na tehniku 2.1 uočljiva je na čvoru br.1. Ovde se zbog zbog manjeg broja praznih čvorova u levom podstablu vrši kretanje u levo, ka čvoru br.2. Na čvoru br.2 kretanje vršimo metodom

slučajnog izbora obzirom da su dubina i broj praznih čvorova u oba smera isti. Prema tome, list čvor koji premeštamo biće ili čvor br.4 ili čvor br.5.

Tehnika 4.1 – Kod ove tehnike, osnovni kriterijum za kretanje kroz stablo u cilju pronalaženja list čvora je veći broj čvorova. Ukoliko je broj čvorova isti u levom i u desnom podstablu, kriterijum za kretanje u levo ili u desno jeste veći broj praznih čvorova. Ako je i u levom i u desnom podstablu broj čvorova i broj praznih čvorova isti, odluka o kretanju u levo ili u desno donosi se metodom slučajnog izbora. Na slici 3.6 prikazan je karakterističan primer za ovu tehniku formiranja stabla sa prioritetima. Na čvoru br.1 vrši se kretanje u desno (na čvor br.3) jer desno podstablo ima veći broj čvorova. Na čvoru br.3 se vrši izbor prema kriterijumu većeg broja praznih čvorova jer je broj čvorova isti. Prema tome, vrši se kretanje u levo ka čvoru br.6. Na čvoru br.6 kretanje vršimo metodom slučajnog izbora obzirom da su brojevi čvorova i brojevi praznih čvorova u oba smera isti. Na kraju, list čvor koji premeštamo biće ili čvor br.8 ili čvor br.9.



Slika 3.6 karakterističan primer stabla za tehnike 4.1 i 4.2

Tehnika 4.2. – Ova tehnika za formiranje stabla sa prioritetima predstavlja izmenjenu verziju tehnike 4.1.. Osnovni kriterijum za kretanje kroz stablo u cilju izbora list čvora je manji broj čvorova. Ovo ujedno predstavlja i jedinu razliku u odnosu na tehniku 4.1.. Kao i kod tehnike 4.1. ukoliko je u levom i desnom podstablu broj čvorova isti, kretanje u levo ili u desno vrši se na osnovu većeg broja praznih čvorova. Ukoliko je i broj praznih čvorova isti u levom i u desnom podstablu, kretanje se vrši metodom slučajnog izbora. Ako ponovo posmatramo primer dat na slici 3.6, razlika u odnosu na tehniku 2.1 uočljiva je na čvoru br.1. Ovde se zbog manjeg broja čvorova u levom podstablu vrši kretanje u levo, ka čvoru br.2. Na čvoru br.2 kretanje vršimo metodom slučajnog izbora obzirom da je broj čvorova i broj praznih čvorova u oba smera isti. Prema tome, list čvor koji premeštamo biće ili čvor br.4 ili čvor br.5.

Tehnika 5. – Ova tehnika formiranja stabla sa prioritetima zasnovana je na slučajnom izboru. Svi list čvorovi koji imaju barem jednog praznog pretka su kandidati za premeštanje.

Tehnika 6.1. – Osnovni kriterijum kod ove tehnike za formiranje stabla sa prioritetima je veći broj praznih predaka. Ova tehnika se razlikuje od tehnika opisanih do sada po tome što se izbor list čvora ne vrši jednim prolaskom kroz stablo od korena do list čvora donošenjem jedne odluke na svakom nivou stabla. Kod ove tehnike potrebno je proći kroz sve čvorove stabla i izabrati list čvor sa najviše praznih predaka. Ukoliko postoji više list čvorova sa istim brojem praznih predaka, bira se čvor veće dubine. Ukoliko je i broj praznih predaka i dubina jednaka kod više posmatranih list čvorova, među njima se bira poslednji „posećeni“ čvor, odnosno čvor koji se nalazi više desno u stablu. Na stablu prikazanom na slici 3.5 izbor postoje 4 čvora sa istim brojem praznih predaka.

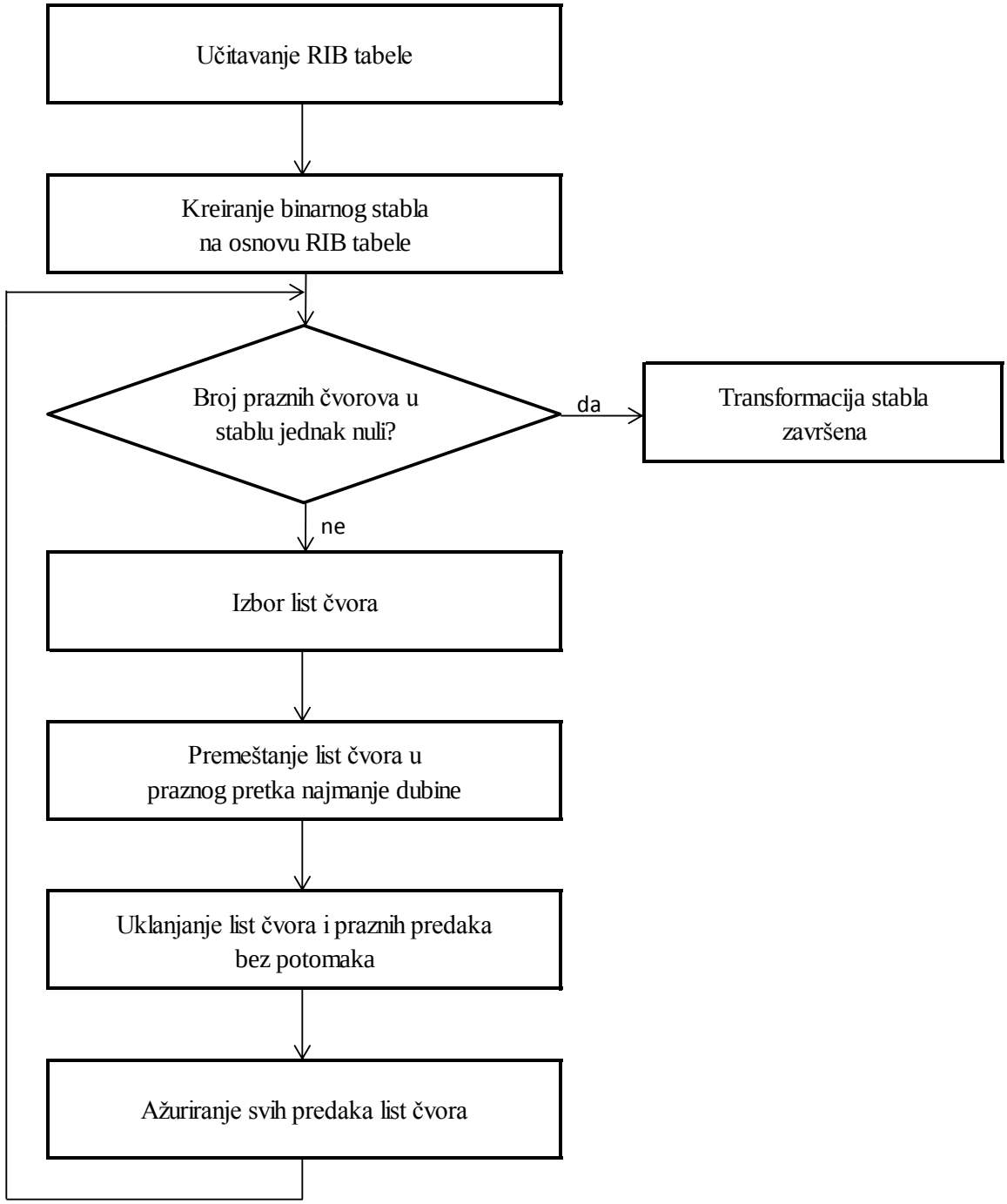
Čvorovi br.11, br.12, br.9 i br.10 imaju 3 prazna pretka. Međutim, čvorovi br.9 i br.10 neće biti izabrani jer čvorovi br.11 i br.12 imaju veću dubinu. Konačno, list čvor koji će biti premešten je čvor br.12 jer će to biti poslednji posećeni čvor, odnosno nalazi se desno u odnosu na čvor br.11.

Tehnika 6.2. – Osnovni kriterijum kod ove tehnike za formiranje stabla sa prioritetima je manji broj praznih predaka. Ovo je jedina razlika u odnosu na tehniku 6.1.. Kao i kod tehnike 6.1., potrebno je proći kroz sve čvorove stabla kako bi se izvršio izbor list čvora. Ukoliko postoji više list čvorova sa istim brojem praznih predaka, bira se list čvor veće dubine. Ukoliko je i broj praznih predaka i dubina jednaka kod više posmatranih list čvorova, bira se čvor koji se nalazi više desno u stablu. Na primeru stabla sa slike 3.5, čvorovi br.4 i br.5 imaju isti broj praznih predaka (2), međutim čvor koji će biti premešten je čvor br.5 jer se nalazi desno u odnosu na čvor br.4.

3.1. Opis kodova tehnika za formiranje stabla sa prioritetima

Tehnike za formiranje stabla sa prioritetima biće implementirane u programskom jeziku C. Za potrebe realizacije koristiće se softverski alat „Code::Blocks“ sa GCC kompajlerom.[3] Kompletni kodovi svih 11 tehnika za formiranje stabla sa prioritetima priloženi su uz tezu na CD-u.

Sve navedene tehnike formiranja stabla sa prioritetima su tehnike transformacije binarnog stabla u stablo sa prioritetima. Na slici 3.1.1 prikazan je algoritam na kome je zasnovano svih 11 navedenih tehnika formiranja stabla sa prioritetima.



Slika 3.1.1 Algoritam za formiranje stabla sa prioritetima

Deo koda koji vrši učitavanje RIB tabele i kreiranje binarnog stabla preuzet je iz rada [4]. U ovom delu rada, biće opisan deo koda koji odgovara preostalim fazama formiranja stabla sa prioritetima, odnosno deo koda koji se tiče transformacije binarnog stabla u stablo sa prioritetima.

Za formiranje binarnog stabla potrebno je kreirati novi tip podataka koji sadrži nekoliko promenljivih, ali i pokazivače. Novi tip podataka se u programskom jeziku C definiše pomoću „*struct*“ funkcije. Na slici 3.1.2 prikazana je definicija novog tipa podataka koja se koristi za formiranje stabla u priloženim kodovima za formiranje stabla sa prioritetima.

```
//struktura za stablo
typedef struct pokstablo
{
  unsigned int prazan; // da li je cvor prazan ili ne
  unsigned int level; // nivo cvora
  unsigned int nl,nd; // broj dece s leve i desne strane
  unsigned int pl,pd; // broj popunjene dece s leve i desne strane
  unsigned int dd,dl; // dubina sa desne i leve strane
  unsigned int prpreci; // broj praznih predaka
  struct pokstablo *levo,*desno,*gore; // pokazivaci za kretanje kroz stablo
}stablo;
```

Slika 3.1.2 *Struct* konstrukcija

Promenljiva „prazan“ predstavlja binarni indikator da li je u pitanju prazan čvor. Ukoliko promenljiva „prazan“ ima vrednost 1, čvor je prazan, ukoliko ima vrednost 0, čvor je popunjen. Promenljiva „level“ sadrži vrednost nivoa na kom se čvor nalazi. Promenljiva „nl“ sadrži vrednost broja čvorova (potomaka) u levom podstablu, dok promenljiva „nd“ sadrži vrednost broja čvorova u desnom podstablu. Promenljiva „pl“ sadrži vrednost broja popunjenih čvorova u levom podstablu, dok promenljiva „pd“ sadrži vrednost broja popunjenih čvorova u desnom podstablu. Vrednosti „dl“ i „dd“ predstavljaju vrednosti dubine levog i desnog podstabla respektivno. Promenljiva „prpreci“ sadrži vrednost broja praznih predaka, i koristi se u tehnikama 6.1. i 6.2.. Pokazivači „levo“, „desno“ i „gore“ predstavljaju pokazivače za kretanje kroz stablo.

Za potrebe analize predloženih tehnika, u kodovima svih 11 tehnika, kreirana je rekurzivna funkcija „statistika“. Ova funkcija poziva se pre i nakon transformacije binarnog stabla u stablo sa prioritetima. Na slici 3.1.3 prikazana je rekurzivna funkcija „statistika“.

```
void statistika (struct pokstablo *p)
{
if (p->dd==p->dl) simdub++;
if ((p->nl==1)&&(p->nd==1)) simlist++;
if (((p->nl==1)&&(p->nd!=1))||((p->nd==1)&&(p->nl!=1))) asimlist++;
if (((p->nl==1)&&(p->desno==NULL))||((p->nd==1)&&(p->levo==NULL))) jedanlist++;
nivo[p->level]++;
if ((p->prazan==0)) pop[p->level]++;
if((p->desno!=NULL)&&(p->levo!=NULL)) simdeca++;
else if((p->desno!=NULL)||(p->levo!=NULL)) asimdeca++;
    else listdeca++;
if (p->levo!=NULL) statistika(p->levo);
if (p->desno!=NULL) statistika(p->desno);
}
```

Slika 3.1.3 – Rekurzivna funkcija „statistika“

Promenljiva „simdub“ sadrži vrednost broja čvorova koji imaju jednaku dubinu u levom i u desnom podstablu. Promenljiva „simlist“ sadrži vrednost broja čvorova koji imaju list čvor i sa leve i sa desne strane. Promenljiva „asimlist“ sadrži vrednost broja čvorova koji imaju jedan list čvor sa leve ili sa desne strane a da pri tome sa druge strane nemaju list čvor. Promenljiva „jedanlist“ sadrži vrednost broja čvorova koji imaju samo jednog potomka koji predstavlja list čvor. Niz „nivo[33]” sadrži vrednosti broja čvorova po nivoima u stablu, pri čemu „nivo[i]” sadrži vrednost broja čvorova u nivou i. Niz „pop[33]” sadrži vrednosti broja popunjenih čvorova po nivoima u stablu. Ovaj niz je od značaja samo za binarno stablo, obzirom da stablo sa prioritetima sadrži samo popunjene čvorove. Promenljiva „listdeca” sadrži broj listova u stablu. Promenljiva „asimdeca” sadrži broj čvorova koji imaju samo jedan aktivan pokazivač (jedan pokazivač je NULL). Promenljiva „simdeca” sadrži broj čvorova koji imaju oba aktivna pokazivača (oba su različita od

NULL). U okviru rekurzivne funkcije “statistika”, sve navedene promenljive se inkrementiraju prema odgovarajućim uslovima, sa izuzetkom niza „nivo[i]” koji se bezuslovno inkrementira. Na kraju funkcije “statistika” vrši se pozivanje iste za levi i za desni čvor ukoliko postoje. Ovime se kreira rekuzija, a funkcija se okončava nakon prolaska kroz sve čvorove u stablu. Pre pokretanja funkcije „statistika“ potrebno je izvršiti inicijalizaciju promenljivih koje se koriste u njoj („simdub“, „simlist“, „asimlist“, „jedanlist“, „listdeca”, „simdeca” , „asimdeca” i „nivo[i]”). Funkcija se poziva sa pokazivačem na koren stabla kao ulaznim parametrom.

3.1.1. Opis koda tehnike 1.1.

Tehnika 1.1. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veća dubina, veći broj praznih čvorova, deterministički (levo). Na slici 3.1.1.1 prikazan je uprošćen deo koda koji se tiče transformacije binarnog stabla u stablo sa prioritetima. Promenljiva „bopt“ sadrži vrednost broja optimizacija, te joj inicijalno dodeljujemo vrednost nula. U ovom radu pod terminom optimizacija smatramo premeštanje jednog list čvora. Pokazivač „niz” pokazuje na koren stabla. Prva *while* petlja je petlja koja se okončava nakon eliminisanja svih praznih čvorova. Ovime je transformacija binarnog stabla završena i formirano je stablo sa prioritetima. U okviru prve *while* petlje prvo se vrši inicijalizacija pokazivača „tren” koji služi za kretanje kroz stablo, i promenljive „brpr“ koja ima vrednost broja praznih čvorova kroz koje je pokazivač „tren“ prošao na putu do list čvora. Nakon toga, ulazi se u *while* petlju koja vrši izbor list čvora prema odgovarajućem kriterijumu. Na izlasku iz petlje „tren“ pokazivač pokazuje na list čvor odnosno čvor koji premeštamo. Sledeća *while* petlja (u okviru prve *while* petlje) vrši uklanjanje list čvora i praznih predaka list čvora koji nemaju druge potomke. Ova *while* petlja takođe vrši ažuriranje vrednosti čvorova od list čvora do korena stabla. Sledeći korak je unošenje prefiksa list čvora u praznog pretka najmanje dubine, odnosno izmenu binarnog indikatora „prazan“. Poslednja *while* petlja (u okviru prve *while* petlje) takođe vrši ažuriranje vrednosti čvorova, ali počev od čvora u koji smeštamo prefiks uklonjenog list čvora do korena stabla.

```
bopt=0; //broj optimizacija
while ((niz->nd)+(niz->nl)-(niz->pd)-(niz->pl)) //dok ima praznih cvorova
    {
    tren=niz;
    brpr=0;
    while((tren->desno)||(tren->levo)) // Slika 3.1.1.2
        {
        // izbor list cvora
        }
    bopt++; //inkrementiranje broj optimizacija
    no=0; //broj uklonjenih cvorova
    while (tren->gore) // Slika 3.1.1.3
        {
        // uklanjanje list cvora i praznih cvorova bez potomaka
        // azuriranje cvorova od list cvora do korena
        }
        novi->prazan=0; //unosenje prefiksa u novi cvor
        tren=novi;
    while (tren->gore) // Slika 3.1.1.4
        {
        // azuriranje cvorova od unetog do korena
        }
    }
```

Slika 3.1.1.1 Uprošćen deo koda koji vrši transformaciju stabla

U nastavku ćemo opisati navedene delove koda. Na slici 3.1.1.2 prikazan je deo koda koji vrši izbor list čvora.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu
        if(tren->dd==tren->dl)//ista dubina levo i desno
            {
            if(((tren->nd)-(tren->pd))==((tren->nl)-(tren->pl))) //ako isti broj
praznih cvorova
                {
                if (((tren->nd)-(tren->pd)==0)&&(brpr==0)) //ako ni levo ni
desno nema praznih cvorova, a nismo prosli ni kroz jedan prazan (nikada ne bi
trebalo da bude ispunjeno)
                    {
                    printf("\n greska 1, nivo: %d\n\n",tren->level);
                    goto izlaz;
                    }
                tren=tren->levo; //u ovom slucaju nije neophodno zbog uslova
ispod
                }
            else if(((tren->nd)-(tren->pd))>((tren->nl)-(tren->pl)))//idi gde
ima vise praznih cvorova
                    {
                    tren=tren->desno;
                    }
                else
                    {
                    tren=tren->levo;
                    }
            }
        else if (brpr==0)
            {
            if ((tren->nd>tren->pd)&&(tren->nl>tren->pl))
            {
                if ((tren->dd>tren->dl))
                    {
                    tren=tren->desno;
                    }
                    else
                    {
                    tren=tren->levo;
                    }
            }
            else if (tren->nd>tren->pd)
                {
                tren=tren->desno;
                }
                else if (tren->nl>tren->pl)
                    {
                    tren=tren->levo;
                    }
                    else //nema praznih ni levo ni desno, niikada ne bi trebalo
da bude ispunjeno
                        {
                        printf ("\n greska 2, nivo: %d\n\n",tren->level);
                        goto izlaz;
                        }
            }//kada u prolazu imamo barem jedan prazan cvor, kriterijum je samo
```

```
veca dubina
            else if (tren->dd>tren->dl)
                    {
                    tren=tren->desno;
                    }
                    else
                    {
                    tren=tren->levo;
                    }
        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.1.2 Deo koda koji vrši izbor list čvora

While petlja u ovom delu koda okončava se pronalaženjem list čvora koji odgovara navedenim kriterijumima. Po ulasku u petlju, vrši se provera da li je čvor na kome se nalazimo prazan, ukoliko jeste vrši se inkrementiranje broja praznih čvorova (na putu do list čvora). Kretanje kroz stablo vrši se na osnovu kriterijuma, pri čemu prvo vršimo proveru da li su ispunjeni preduslovi za kriterijume manjeg prioriteta. Prema tome, prvo vršimo proveru da li je dubina levog i desnog podstabla jednaka, a zatim da li je broj praznih čvorova u levom i u desnom stablu jednak. Ukoliko su oba uslova ispunjena izbor se vrši na osnovu kriterijuma najmanjeg prioriteta i vršimo kretanje u levo. Ukoliko broj praznih čvorova nije jednak u levom i u desnom podstablu, a ispunjen je uslov da je dubina jednaka u oba podstabla, vršimo kretanje u smeru u kome ima više praznih čvorova. Kako se kretanje vrši u smeru u kome ima više praznih čvorova, nije potrebna vršiti proveru da li smo prošli kroz barem jedan prazan čvor (na putu do list čvora). Ukoliko dubina i broj praznih čvorova nemaju iste vrednosti u levom i u desnom podstablu, vršimo proveru da li je broj praznih čvorova koje smo prošli (na putu do list čvora) jednak nuli. Ukoliko jeste, ne smemo se kretati u smeru u kome nema praznih čvorova. Ukoliko je broj praznih čvorova u oba smera veći od nule, kretanje se vrši na osnovu primarnog kriterijuma - veće dubine. Ukoliko broj praznih čvorova nije veći od nule u oba smera, kretanje se vrši u pravcu u kome ima praznih čvorova. Ukoliko broj praznih čvorova koje smo prošli nema vrednost nula, to znači da postoji barem jedan čvor u koji možemo premestiti prefiks list čvora, i možemo vršiti kretanje na osnovu primarnog kriterijuma - veće dubine. Na kraju *while* petlje vrednost pokazivača „opt“ dobija vrednost pokazivača „tren“ odnosno pokazivača na kome se nalazimo. Na taj način, po okončavanju petlje, pokazivač „opt“ pokazivaće na list čvor koji premeštamo.

Na slici 3.1.1.3 prikazan je deo koda koji vrši uklanjanje list čvora, praznih predaka list čvora koji nemaju druge potomke i ažuriranje vrednosti čvorova od list čvora do korena stabla. Pre ulaska u *while* petlju, pokazivač „tren“ pokazuje na list čvor čiji prefiks premeštamo. U okviru *while* petlje, vrši se kretanje na gore i petlja se okončava po dolasku do korena stabla. Pokazivač „sled“ predstavlja pokazivač na čvor koji se nalazi iznad čvora na kome se nalazimo. Nakon okončavanja petlje, pokazivač „novi“ pokazivaće na praznog potomka najmanje dubine u koji smeštamo prefiks list čvora. Ovo postižemo tako što na početku petlje vršimo proveru da li je čvor na kome se nalazimo prazan. Ukoliko jeste, pokazivač „novi“ dobija vrednost pokazivača na čvor na kome se nalazimo („tren“ pokazivač). Kako uslov petlje nije ispunjen kada se „tren“ pokazivač nalazi na korenu stabla, ukoliko je koren prazan, pokazivaču „novi“ dodeljuje se vrednost pokazivača na koren stabla („niz“ pokazivač).

Sledeći korak je provera da li je potrebno ukloniti čvor na kome se nalazimo. U okviru uslova za uklanjanje čvora razlikujemo dva slučaja. Jedan slučaj je slučaj u kome čvor na kome se nalazimo predstavlja list čvor čiji prefiks premeštamo. Drugi slučaj je slučaj u kome čvor na kome se nalazimo predstavlja prazan čvor koji nema potomke, a pri tome nije jedini prazan čvor u putanji

do list čvora. Ukoliko je uslov za uklanjanje čvora ispunjen, promenljivu „no“ koja sadrži vrednost broja uklonjenih čvorova uvećavamo za jedan. Zatim, ukoliko je čvor koji uklanjamo prazan, vrednost broja praznih čvorova „brpr“ umanjujemo za jedan. Pre samog uklanjanja čvora, moramo ažurirati vrednosti čvora koji se nalazi iznad čvora koji uklanjamo. Na čvoru koji se nalazi iznad, brišemo pokazivač na čvor koji uklanjamo, a vrednostima broja čvorova, broja popunjenih čvorova i dubine (u pravcu čvora koji uklanjamo) dodeljujemo vrednost nula. Prilikom ažuriranja vrednosti dubine levo i dubine desno kod predaka, potrebno je uzeti u obzir činjenicu da uklanjanje „no“ čvorova ne znači nužno i smanjivanje dubine u svim precima za „no“. Zbog ovoga, uvodimo promenljivu „ddd“ koja će kasnije biti ključna za ažuriranje vrednosti dubina. Promenljivoj „ddd“ dodeljujemo vrednost dubine (levo ili desno) čvora iznad čvora koji uklanjamo, umanjenu za broj uklonjenih čvorova. Na kraju dela koda koji se tiče uklanjanja čvora, vršimo oslobađanje memorije alocirane za čvor koji uklanjamo.

Ukoliko nije ispunjen uslov za uklanjanje čvora, potrebno je ažurirati vrednosti čvora iznad čvora na kome se nalazimo. Vrednosti se ažuriraju na osnovu postojećih vrednosti, broja uklonjenih čvorova („no“) i promenljive „ddd“. Broj čvorova umanjuje se za broj uklonjenih čvorova. Broj popunjenih čvorova umanjuje se za jedan. Za ažuriranje vrednosti dubina (levo i desno) ključnu ulogu ima promenljiva „ddd“ koju je takođe potrebno ažurirati. Vrednost promenljive „ddd“ se ažurira tako što joj se dodeljuje vrednost dubine levo ili dubine desno čvora na kome se nalazimo, ukoliko je veća. Ukoliko nije veća, vrednost promenljive „ddd“ ostaje nepromenjena. Sada, čvoru iznad čvora na kome se nalazimo možemo ažurirati vrednost dubine, dodeljivanjem vrednosti „ddd“. Zatim, vrednost „ddd“ se ažurira prema maksimalnoj vrednosti dubine čvora iznad čvora na kome se nalazimo. Ovo je bitno za ažuriranje u narednim iteracijama. Na kraju petlje, vrši se kretanje ka gore.

```
while (tren->gore)
    {
    sled=tren->gore;
    if (tren->prazan==1) novi=tren; //nakon prolaska, cvor u koji smestamo list
ce biti prazan cvor najmanje dubine
    if (niz->prazan==1) novi=niz; //uslov (tren->gore) petlje ne ispituje koren
stabla
    if (((tren->desno==NULL)&&(tren->levo==NULL)&&(tren-
>prazan==1)&&(brpr>1))||(tren==opt)) //ako je u pitanju prazan cvor koji nema
decu (u ovom slucaju je bitno da nije poslednji prazan cvor u putanji), ili ako
je u pitanju cvor sa prefiksom koji se izmesta
        {
        no++;
        if (tren->prazan==1) brpr--;
        if (sled->desno==tren) //azuriranje
            {
            sled->desno=NULL;
            sled->nd=0;
            sled->pd=0;
            ddd=sled->dd-no; // potrebno za azuriranje dubine u cvorovima
            sled->dd=0;
            }
        if(sled->levo==tren) //azuriranje
            {
            sled->levo=NULL;
            sled->nl=0;
            sled->pl=0;
            ddd=sled->dl-no;
            sled->dl=0;
            }
```

```
        free(tren);
        }
  //azuriranje ostalih cvorova
    else
    {
        if (sled->desno==tren)
        {
        sled->nd=sled->nd-no;
        sled->pd--;
        if (tren->dl>ddd) ddd=tren->dl;
        if (tren->dd>ddd) ddd=tren->dd;
        sled->dd=ddd;
        if (sled->dl>sled->dd) ddd=sled->dl;
        }
        if (sled->levo==tren)
        {
        sled->nl=sled->nl-no;
        sled->pl--;
        if (tren->dl>ddd) ddd=tren->dl;
        if (tren->dd>ddd) ddd=tren->dd;
        sled->dl=ddd;
        if (sled->dd>sled->dl) ddd=sled->dd;
        }
    }
    tren=sled; //kretanje na gore
    }
```

Slika 3.1.1.3 Deo koda koji vrši ažuriranje i uklanjanje čvorova

Na slici 3.1.1.4 prikazan je deo koda koji vrši ažuriranje čvorova od čvora u koji je unet prefiks, do korena stabla. Kada unesemo prefiks u praznog pretka najmanje dubine, ostaje nam da u čvorovima iznad tog čvora uvećamo vrednost broja popunjenih čvorova, koju smo smanjili u prethodnoj *while* petlji. Na taj način, ukupan broj popunjenih čvorova u stablu ostaje nepromenjen.

```
    while (tren->gore) //azuriranje cvorova od unetog na gore
        {
        if (tren->gore->desno==tren)
            {
            tren->gore->pd++;
            }
            else tren->gore->pl++;
        tren=tren->gore;
        }
```

Slika 3.1.1.4 Deo koda koji vrši dodatno ažuriranje čvorova

Obzirom da deo koda koji vrši uklanjanje i ažuriranje čvorova nije izmenjen za potrebe implemetacije drugih tehnika formiranja stabla sa prioritetima, prilikom opisa koda preostalih tehnika, biće opisan samo deo koda koji vrši izbor list čvora koji se premešta.

3.1.2. Opis koda tehnike 1.2.

Tehnika 1.2. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veća dubina, veći broj praznih čvorova, statistički (slučajan izbor). Na slici 3.1.2.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Obzirom da se ova tehnika razlikuje od tehnike 1.1. samo u trećem kriterijumu izbora list čvora, razlika u kodu je mala u odnosu na kod tehnike 1.1.. Jedina razlika u odnosu na kod tehnike 1.1. nalazi se u delu koda gde se vrši provera da li je dubina levog i desnog podstabla jednaka i da li je broj praznih čvorova u levom i u desnom stablu jednak. Ukoliko su oba uslova ispunjena izbor se vrši na osnovu kriterijuma najmanjeg prioriteta i kretanje se vrši slučajnim izborom. Ovo je realizovano na sledeći način. Uvedena je nova promenljiva „rndbr“ kojoj se dodeljuje vrednost funkcije *rand* (*rndbr=rand()%2*). Ovime se promenljivoj „rndbr“ dodeljuje slučajna celobrojna vrednost između 0 i 1 (odnosno vrednost 0 ili 1). Nakon toga, kretanje se vrši u levo ukoliko je vrednost promenljive „rndbr“ jednaka nuli, ili u desno ako je vrednost promenljive „rndbr“ jednaka jedinici. U kodovima preostalih tehnika formiranja stabla sa prioritetima, kretanje na osnovu slučajnog izbora biće realizovano na isti način.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu

        if(tren->dd==tren->dl)//ista dubina levo i desno
            {
            if(((tren->nd)-(tren->pd))==((tren->nl)-(tren->pl))) //ako isti broj
praznih cvorova
                {
                if (((tren->nd)-(tren->pd)==0)&&(brpr==0)) //ako ni levo ni
desno nema praznih cvorova, a nismo prosli ni kroz jedan prazan (niikada ne bi
trebalo da bude ispunjeno)
                    {
                    printf("\n greska 1, nivo: %d\n\n",tren->level);
                    goto izlaz;
                    }

                rndbr=rand()%2; // slucajan izbor (0 ili 1)
                if (rndbr==0) tren=tren->levo;
                    else tren=tren->desno;
                }
            else if(((tren->nd)-(tren->pd))>((tren->nl)-(tren->pl)))//idi gde
ima vise praznih cvorova
                    {
                    tren=tren->desno;
                    }
                else
                    {
                    tren=tren->levo;
                    }
            }
        else if (brpr==0)
            {
            if ((tren->nd>tren->pd)&&(tren->nl>tren->pl))
            {
                if ((tren->dd>tren->dl))
                    {
                    tren=tren->desno;
                    }
```

```
                    else
                    {
                    tren=tren->levo;
                    }
            }
            else if (tren->nd>tren->pd)
                {
                tren=tren->desno;
                }
                else if (tren->nl>tren->pl)
                    {
                    tren=tren->levo;
                    }
                    else //nema praznih ni levo ni desno, niikada ne bi trebalo
da bude ispunjeno
                        {
                        printf ("\n greska 2, nivo: %d\n\n",tren->level);
                        goto izlaz;
                        }
            }
            //kada u prolazu imamo barem jedan prazan cvor, kriterijum je samo
veca dubina
            else if (tren->dd>tren->dl)
                    {
                    tren=tren->desno;
                    }
                    else
                    {
                    tren=tren->levo;
                    }
        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.2.1 Deo koda koji vrši izbor list čvora

3.1.3. Opis koda tehnike 2.1.

Tehnika 2.1. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veća dubina, veći broj čvorova, slučajan izbor. Na slici 3.1.3.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Po ulasku u petlju, vrši se provera da li je čvor na kome se nalazimo prazan, i ukoliko jeste vrši se inkrementiranje broja praznih čvorova (na putu do list čvora). Kretanje kroz stablo vrši se na osnovu kriterijuma, pri čemu prvo vršimo proveru da li su ispunjeni preduslovi za kriterijume manjeg prioriteta. Prema tome, prvo vršimo proveru da li je dubina levog i desnog podstabla jednaka, a zatim da li je broj čvorova u levom i u desnom stablu jednak. Ukoliko je dubina levog i desnog podstabla jednaka, vršimo proveru da li je broj praznih čvorova koje smo prošli (na putu do list čvora) jednak nuli. Ukoliko jeste, ne smemo se kretati u smeru u kome nema barem jedan prazan čvor. Zbog toga vršimo proveru da li u oba smera ima praznih čvorova. Ukoliko ima, prvo vršimo proveru da li je jednak broj čvorova levo i desno, i ukoliko jeste kretanje se vrši prema poslednjem kriterijumu – metodom slučajnog izbora. Zatim, ukoliko nije isti broj čvorova levo i desno (a postoje prazni čvorovi levo i desno) kretanje se vrši prema drugom kriterijumu, odnosno prema većem broju čvorova. Ukoliko broj praznih čvorova nije veći od nule u oba smera, kretanje se vrši u pravcu gde je broj praznih čvorova veći od nule. Na ovaj način obezbeđuje se

barem jedan prazan čvor za smeštanje list čvora. Ukoliko broj praznih čvorova koje smo prošli nije jednak nuli, a zadovoljen je uslov da je dubina jednaka levo i desno, vršimo proveru da li je broj čvorova jednak levo i desno. Ukoliko jeste, kretanje se vrši na osnovu trećeg kriterijuma – metodom slučajnog izbora. Ukoliko dubina levo nije jednaka dubini desno, kretanje se vrši prema primarnom kriterijumu, odnosno prema većoj dubini. Prvo proveravamo da li je broj praznih čvorova koji smo prošli jednak nuli. Ukoliko jeste, ponovo vršimo proveru da li je broj praznih čvorova veći i levo i desno. Ukoliko jeste, kretanje se vrši prema primarnom kriterijumu – većoj dubini. Ukoliko broj praznih čvorova nije veći od nule u oba smera, moramo se kretati u smeru u kome ima barem jedan prazan čvor. Ukoliko je broj praznih čvorova koje smo prošli veći od nule, kretanje se vrši prema primarnom kriterijumu – većoj dubini. Na kraju *while* petlje vrednost pokazivača „opt“ dobija vrednost pokazivača „tren“ odnosno pokazivača na kome se nalazimo. Na taj način, po okončavanju petlje, pokazivač „opt“ pokazivaće na list čvor koji premeštamo.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu

        if(tren->dd==tren->dl) //ista dubina levo i desno
            {
            if (brpr==0) //ako nema praznih potrebno je ukljuciti i uslov da u
putanji ima praznih cvorova
                {
                if ((tren->nd>tren->pd)&&(tren->nl>tren->pl)) //ako i levo i
desno ima praznih
                    {
                    if (tren->nl==tren->nd) //uslov ako je i dubina i broj
cvorova isti
                        {
                        rndbr=rand()%2;
                        if (rndbr==0) tren=tren->levo;
                        else tren=tren->desno;
                        }
                    else if (tren->nl>tren->nd) tren=tren->levo;
                        else tren=tren->desno;
                    }
                else if (tren->nl>tren->pl) tren=tren->levo; //ide gde ima
praznih
                    else tren=tren->desno;
                }
            else if (tren->nl==tren->nd) //uslov ako je i dubina i broj cvorova
isti
                        {
                        rndbr=rand()%2;
                        if (rndbr==0) tren=tren->levo;
                        else tren=tren->desno;
                        }
                else if (tren->nl>tren->nd) tren=tren->levo;
                    else tren=tren->desno;
            }
        else if (brpr==0)
            {
            if ((tren->nd>tren->pd)&&(tren->nl>tren->pl))
            {
                if ((tren->dd>tren->dl))  tren=tren->desno;
                    else tren=tren->levo;
            }
            else if (tren->nd>tren->pd) tren=tren->desno;
```

```
                else if (tren->nl>tren->pl) tren=tren->levo;

                    else //nema praznih ni levo ni desno, niikada ne bi trebalo
da bude ispunjeno
                        {
                        printf ("\n greska 2, nivo: %d\n\n",tren->level);
                        printf ("\n nivo: %d\n brpr: %d\n dub.levo: %d\n
dub.desno: %d\n br.cv.levo: %d\n br.cv.desno: %d\n br.pcv.levo: %d\n
br.pcv.desno: %d\n",tren->level,brpr,tren->dl,tren->dd,tren->nl,tren->nd,tren-
>pl,tren->pd);
                        goto izlaz;
                        }
            }
            //kada u prolazu imamo barem jedan prazan cvor, kriterijum je samo
veca dubina
            else if (tren->dd>tren->dl) tren=tren->desno;
                 else tren=tren->levo;

        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.3.1 Deo koda koji vrši izbor list čvora

3.1.4. Opis koda tehnike 2.2.

Tehnika 2.2. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veća dubina, manji broj čvorova, slučajan izbor. Na slici 3.1.4.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Obzirom da se ova tehnika razlikuje od tehnike 2.1. samo u drugom kriterijumu izbora list čvora, razlika u kodu je mala u odnosu na kod tehnike 2.1.. Jedina razlika u odnosu na kod tehnike 2.1. nalazi se u delu koda gde se vrši provera da li je dubina levog i desnog podstabla jednaka. Ukoliko jeste, kretanje se vrši u smeru u kome ima manje čvorova.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu

        if(tren->dd==tren->dl) //ista dubina levo i desno
            {
            if (brpr==0) // ako nema praznih potrebno je ukljuciti i uslov da u
putanji ima praznih cvorova
                {
                if ((tren->nd>tren->pd)&&(tren->nl>tren->pl)) //ako i levo i
desno ima praznih
                    {
                    if (tren->nl==tren->nd) //uslov ako je i dubina i broj
cvorova isti
                        {
                        rndbr=rand()%2;
                        if (rndbr==0) tren=tren->levo;
                        else tren=tren->desno;
                        }
                    else if (tren->nl<tren->nd) tren=tren->levo;
                        else tren=tren->desno;
                    }
                else if (tren->nl>tren->pl) tren=tren->levo; //ako na obe strane
```

```
nema praznih, ide gde ima
                    else tren=tren->desno;
                }
            else if (tren->nl==tren->nd)  //uslov ako je i dubina i broj cvorova
isti
                        {
                        rndbr=rand()%2;
                        if (rndbr==0) tren=tren->levo;
                        else tren=tren->desno;
                        }
                else if (tren->nl<tren->nd) tren=tren->levo;
                    else tren=tren->desno;
            }

        else if (brpr==0)
            {
            if ((tren->nd>tren->pd)&&(tren->nl>tren->pl))
            {
                if ((tren->dd>tren->dl)) tren=tren->desno;
                else tren=tren->levo;
            }
            else if (tren->nd>tren->pd) tren=tren->desno;
                else if (tren->nl>tren->pl) tren=tren->levo;
                    else //nema praznih ni levo ni desno, niikada ne bi trebalo
da bude ispunjeno
                        {
                        printf ("\n greska 2, nivo: %d\n\n",tren->level);
                        printf ("\n nivo: %d\n brpr: %d\n dub.levo: %d\n
dub.desno: %d\n br.cv.levo: %d\n br.cv.desno: %d\n br.pcv.levo: %d\n
br.pcv.desno: %d\n",tren->level,brpr,tren->dl,tren->dd,tren->nl,tren->nd,tren-
>pl,tren->pd);
                        goto izlaz;
                        }
            }
            //kada u prolazu imamo barem jedan prazan cvor, kriterijum je samo
veca dubina
            else if (tren->dd>tren->dl) tren=tren->desno;
                else tren=tren->levo;

        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.4.1 Deo koda koji vrši izbor list čvora

3.1.5. Opis koda tehnike 3.1.

Tehnika 3.1. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veći broj praznih čvorova, veća dubina, slučajan izbor. Na slici 3.1.5.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Po ulasku u petlju, vrši se provera da li je čvor na kome se nalazimo prazan, ukoliko jeste vrši se inkrementiranje broja praznih čvorova (na putu do list čvora). U ovom kodu uvedene su dve nove promenljive: „prazl“ i „prazd“. Ove promenljive predstavljaju broj praznih čvorova u levom i u desnom podstablu respektivno, i dodeljuju im se vrednosti razlike broja čvorova i broja praznih čvorova u levom i desnom podstablu. Kao i kod prethodno opisanih kodova, prvo vršimo proveru da li su ispunjeni preduslovi za kriterijume manjeg prioriteta. Ukoliko je broj

praznih čvorova u levom i desnom podstablu jednak, vršimo proveru da li je dubina levo i desto jednaka. Ukoliko su oba uslova zadovoljena, kretanje vršimo prema trećem uslovu – metodom slučajnog izbora. Ukoliko dubina levo i desno nije ista, a broj praznih čvorova jeste, kretanje se vrši u pravcu veće dubine. Na kraju, ukoliko broj praznih čvorova nije isti, kretanje se vrši u pravcu u kome ima više praznih čvorova. U ovom kodu, nije vršena provera da li postoji barem jedan prazan čvor u putanji do list čvora zato što je primarni kriterijum ove tehnike veći broj praznih čvorova. Po okončavanju petlje, pokazivač „opt“ pokazivaće na list čvor koji premeštamo.

```
while((tren->desno)||(tren->levo)) //radi do list cvora
    {
    if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu
    prazl=tren->nl-tren->pl;
    prazd=tren->nd-tren->pd;

    if(prazl==prazd) //isti broj praznih cvorova levo i desno
        {
        if (tren->dl==tren->dd) // ako je isti broj praznih cvorova i dubina
                    {
                    rndbr=rand()%2;
                    if (rndbr==0) tren=tren->levo;
                    else tren=tren->desno;
                    }
        else if (tren->dl>tren->dd) tren=tren->levo;
            else tren=tren->desno;
        }
        else if (prazl>prazd) tren=tren->levo;
            else tren=tren->desno;
    opt=tren;
    } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.5.1 Deo koda koji vrši izbor list čvora

3.1.6. Opis koda tehnike 3.2.

Tehnika 3.2. je tehnika u kojoj se izbor čvora vrši po kriterijumima: manji broj praznih čvorova, veća dubina, slučajan izbor. Na slici 3.1.6.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Ova tehnika razlikuje se od tehnike 3.1. po tome što je primarni kriterijum izbora list čvora manji broj praznih čvorova. Zbog toga, u realizaciji ove tehnike moramo uvesti proveru kojom obezbeđujemo barem jedan prazan čvor u putanji do list čvora. Deo koda koji vrši kretanje na osnovu drugi i trećeg kriterijuma ostaje nepromenjen. Ukoliko broj praznih čvorova nije jednak u levom i u desnom podstablu, vrši se provera da li je proj praznih čvorova koje smo prošli jednak nuli. Ukoliko jeste, ne smemo vršiti kretanje u pravcu u kome nema barem jedan prazan čvor. Ukoliko je broj praznih čvorova veći od nule u oba podstabla, kretanje se vrši na osnovu primarnog kriterijuma – manjeg broja praznih čvorova. Ukoliko nije, kretanje se vrši u pravcu u kome ima barem jedan prazan čvor. Ukoliko broj praznih čvorova koje smo prošli nije jednak nuli, kretanje se vrši na osnovu primarnog kriterijuma – manjeg broja praznih čvorova.

```
while((tren->desno)||(tren->levo)) //radi do list cvora
    {
    if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu
    prazl=tren->nl-tren->pl;
    prazd=tren->nd-tren->pd;
```

```
        if(prazl==prazd)//isti broj praznih cvorova levo i desno
            {
            if (tren->dl==tren->dd) // ako je isti broj praznih cvorova i dubina
                        {
                        rndbr=rand()%2;
                        if (rndbr==0) tren=tren->levo;
                        else tren=tren->desno;
                        }
            else if (tren->dl>tren->dd) tren=tren->levo;
                else tren=tren->desno;
            }
            else if (brpr==0) //ako nema praznih cvorova moramo doci barem do
jednog za smestanje novog cvora
                {
                if ((prazl>0)&&(prazd>0)) //ako i levo i desno ima praznih
cvorova ide gde ih ima manje
                    {
                    if (prazl<prazd) tren=tren->levo;
                    else tren=tren->desno;
                    }
                else if (prazl>0) tren=tren->levo; //ako na jednoj strani nema
praznih cvorova mora ici tamo gde ih ima
                    else tren=tren->desno;
                }
                else if ((tren->nl>0)&&(tren->nd>0)) //(imamo barem jedan
prazan)ako i levo i desno ima cvorova, onda idemo tamo gde ima manje praznih
cvorova
                    {
                    if (prazl<prazd) tren=tren->levo;
                    else tren=tren->desno;
                    }
                    else if (tren->nl>0) tren=tren->levo;
                        else tren=tren->desno;

        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.6.1 Deo koda koji vrši izbor list čvora

3.1.7. Opis koda tehnike 4.1.

Tehnika 4.1. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veći broj čvorova, veći broj praznih čvorova, slučajan izbor. Na slici 3.1.7.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Po ulasku u petlju, vrši se ažuriranje promenljivih „brpr“, „prazl“ i „prazd“. Nakon toga, vrši se provera da li je broj čvorova u levom i desnom podstablu jednaka. Ukoliko jeste, vrši se provera da li je i broj praznih čvorova jednak u oba podstabla. Ukoliko su ova dva uslova ispunjena, kretanje se vrši na osnovu trećeg kriterijuma – metodom slučajnog izbora. Ukoliko broj praznih čvorova u oba podstabla nije jednak, a broj čvorova jeste, vrši se kretanje u pravcu u kome ima više praznih čvorova (drugi kriterijum). Ukoliko broj čvorova nije jednak u oba podstabla, vrši se provera da li je proj praznih čvorova koje smo prošli jednak nuli. Ukoliko jeste, kretanje se vrši u pravcu u kome ima veći broj čvorova (primarni kriterijum) ukoliko u oba podstabla postoji barem jedan prazan čvor. Ukoliko u oba podstabla ne postoji barem po jedan prazan čvor, kretanje se vrši u pravcu u kome postoji barem jedan prazan čvor. Ukoliko broj praznih

čvorova koje smo prošli nije jednak nuli, kretanje se vrši na osnovu primarnog kriterijuma – veći broj čvorova. Po okončavanju petlje, pokazivač „opt“ pokazivaće na list čvor koji premeštamo.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu
        prazl=tren->nl-tren->pl;
        prazd=tren->nd-tren->pd;

        if(tren->nl==tren->nd)//isti broj cvorova levo i desno
            {
            if (prazl==prazd) // ako je isti broj praznih, slucajan izbor
                {
                rndbr=rand()%2; //slucajan broj 0/1
                if (rndbr==0) tren=tren->levo;
                else tren=tren->desno;
                }
            else if (prazl>prazd) tren=tren->levo;  //idi gde ima vise praznih
                else tren=tren->desno;
            }
            else if (brpr==0) //ako nema praznih cvorova moramo doci barem do
jednog za smestanje list cvora
                {
                if ((prazl>0)&&(prazd>0)) //ako i levo i desno ima praznih
cvorova ide gde ima manje
                    {
                    if (tren->nl>tren->nd) tren=tren->levo;
                    else tren=tren->desno;
                    }
                else if (prazl>0) tren=tren->levo; //ide gde ima praznih c.
                    else tren=tren->desno;
                }
                else if (tren->nl>tren->nd) tren=tren->levo; //postoji barem
jedan prazan
                    else tren=tren->desno;

        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.7.1 Deo koda koji vrši izbor list čvora

3.1.8. Opis koda tehnike 4.2.

Tehnika 4.2. je tehnika u kojoj se izbor čvora vrši po kriterijumima: manji broj čvorova, veći broj praznih čvorova, slučajan izbor. Na slici 3.1.8.1 prikazan je deo koda koji vrši izbor list čvora prema ovim kriterijumima. Ova tehnika razlikuje se od tehnike 4.1. po tome što je primarni kriterijum izbora list čvora manji broj čvorova. Zbog toga, u realizaciji ove tehnike moramo uvesti proveru kojom proveravamo da li u oba podstabla postoji barem jedan čvor. Deo koda koji vrši kretanje na osnovu drugog i trećeg kriterijuma ostaje nepromenjen. Ukoliko broj čvorova nije jednak u levom i u desnom podstablu, vrši se provera da li je broj praznih čvorova koje smo prošli jednak nuli. Ukoliko jeste, vrši se provera da li u oba podstabla postoji barem po jedan prazan čvor, i ukoliko postoji, kretanje se vrši na osnovu primarnog kriterijuma – manjeg broja čvorova. Ukoliko

ne postoji po jedan prazan čvor u oba podstabla, kretanje se vrši u pravcu u kome ima barem jedan prazan čvor. Ukoliko broj praznih čvorova koje smo prošli nije jednak nuli, vrši se provera da li je broj čvorova u oba podstabla veći od nule. Ukoliko jeste, smemo se kretati u pravcu u kome ima manje čvorova. Ukoliko broj čvorova nije veći od nule u jednom podstablu, krećemo se u pravcu u kome ima čvorova.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu
        prazl=tren->nl-tren->pl;
        prazd=tren->nd-tren->pd;

        if(tren->nl==tren->nd)//isti broj cvorova levo i desno
            {
            if (prazl==prazd) // ako je isti broj praznih, slucajan izbor
                {
                rndbr=rand()%2; //slucajan broj 0/1
                if (rndbr==0) tren=tren->levo;
                else tren=tren->desno;
                }
            else if (prazl>prazd) tren=tren->levo;  //idi gde ima vise praznih
                else tren=tren->desno;
            }
            else if (brpr==0) //ako nema praznih cvorova moramo doci barem do
jednog za smestanje novog cvora
                {
                if ((prazl>0)&&(prazd>0)) //ako i levo i desno ima praznih
cvorova ide gde ima manje
                    {
                    if (tren->nl<tren->nd) tren=tren->levo;
                    else tren=tren->desno;
                    }
                else if (prazl>0) tren=tren->levo; //ide gde ima praznih c.
                    else tren=tren->desno;
                }
                else if ((tren->nl>0)&&(tren->nd>0))
                    {
                    if (tren->nl<tren->nd) tren=tren->levo; //imamo barem jedan
prazan ide gde ima manje cvorova
                    else tren=tren->desno;
                    }
                    else if (tren->nl>0) tren=tren->levo;
                        else tren=tren->desno;
        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.8.1 Deo koda koji vrši izbor list čvora

3.1.9. Opis koda tehnike 5.

Tehnika 5. je tehnika u kojoj se izbor čvora vrši metodom slučajnog izbora. Na slici 3.1.9.1 prikazan je deo koda koji vrši izbor list čvora metodom slučajnog izbora. Po ulasku u petlju, vrši se ažuriranje promenljivih „brpr“, „prazl“ i „prazd“. Nakon toga, promenljivoj „rndbr“ se dodeljuje vrednost funkcije *rand* (*rndbr=rand()%2*). Ovime se promenljivoj „rndbr“ dodeljuje celobrojna

vrednost 0 ili 1 slučajnim izborom. Nakon toga, vrši se provera da li je broj praznih čvorova koje smo prošli jednak nuli. Ukoliko jeste, kretanje se ne sme vršiti u pravcu u kome ne postoji barem jedan prazan čvor. Prema tome, ukoliko u oba podstabla postoji barem jedan prazan čvor, kretanje se vrši metodom slučajnog izbora na osnovu vrednosti promenljive „rndbr“ (0 – levo, 1 – desno). Ukoliko ne postoji barem jedan prazan čvor u oba podstabla, kretanje se vrši u pravcu u kome postoji barem jedan prazan čvor. Ukoliko je broj čvorova koje smo prošli veći od nule, vrši se provera da li čvor na kome se nalazimo ima barem po jedno dete u oba podstabla. Ukoliko ima, kretanje se vrši metodom slučajnog izbora. Ukoliko nema, kretanje se vrši u pravcu u kome postoji barem jedno dete. Po okončavanju petlje, pokazivač „opt“ pokazivaće na list čvor koji premeštamo.

Pre dela koda koji se tiče transformacije binarnog stabla u stablo sa prioritetima, u kod je dodata komanda *srand(time(NULL));*. Ovom komandom generiše se nova pseudo-slučajna sekvenca, na osnovu koje promenljiva „rndbr“ dobija vrednost. Bez dodavanja ove komande, rezultat koda bio bi isti prilikom svakog pokretanja.

```
    while((tren->desno)||(tren->levo)) //radi do list cvora
        {
        if (tren->prazan==1) brpr++; //brojac praznih cvorova u prolazu
        prazl=tren->nl-tren->pl;
        prazd=tren->nd-tren->pd;

        rndbr=rand()%2; //generise slucajan broj 0 ili 1

        if (brpr==0) //ukoliko nema praznih u prolazu
            {
            if ((prazl>0)&&(prazd>0))  //ako i levo i desno ima praznih -
slucajan izbor
                {
                if (rndbr==0) tren=tren->levo;
                else tren=tren->desno;
                }
            else if (prazl>0) tren=tren->levo;  //ide gde ima praznih ako nema
na obe strane
                else tren=tren->desno;
            }
        else if ((tren->nl>0)&&(tren->nd>0))  //ako ima cvorova i levo i desno
ide nasumicno
            {
            if (rndbr==0) tren=tren->levo;
            else tren=tren->desno;
            }
            else if (tren->nl>0) tren=tren->levo;
                else tren=tren->desno;

        opt=tren;
        } //ovde nam je opt/tren tren cvor koji premestamo
```

Slika 3.1.9.1 Deo koda koji vrši izbor list čvora

3.1.10. Opis koda tehnike 6.1.

Tehnika 6.1. je tehnika u kojoj se izbor čvora vrši po kriterijumima: veći broj praznih predaka, veća dubina. Realizacija ove tehnike razlikuje se od ostalih po tome što se izbor list čvora

ne vrši jednim prolaskom kroz stablo od korena do list čvora donošenjem jedne odluke na svakom nivou stabla. Kod ove tehnike potrebno je proći kroz sve čvorove stabla i izabrati list čvor sa najviše praznih predaka. To se postiže kreiranjem rekurzivne funkcije. Rekurzivna funkcija „praznipreci“ definisana je u delu koda iznad dela koji se tiče transformacije binarnog stabla u stablo sa prioritetima. Na slici 3.1.10.1 prikazan je deo koda u kome je definisana ova rekurzivna funkcija. Ova funkcija vrši ažuriranje vrednosti „prpreci“ u svim čvorovima u stablu, ali vrši i izbor list čvora koji premeštamo. Ulazni parametar je pokazivač „p“, dok je izlazni parametar pokazivač na list čvor koji izmeštamo („opt“). Po ulasku u funkciju, vrši se provera da li pokazivač pokazuje na koren stabla, i ako pokazuje, vrednost „prpreci“ korena stabla dobija vrednost nula. Ukoliko pokazivač ne pokazuje na koren stabla, vrši se upit da li je čvor iznad čvora na kome se nalazimo prazan. Ukoliko jeste, vrednost „prpreci“ čvora na kome se nalazimo inkrementira se za jedan. Ukoliko čvor iznad nije prazan, vrednost „prpreci“ čvora na kome se nalazimo dobija vrednost „prpreci“ čvora iznad čvora na kome se nalazimo (ne inkrementira se). Na ovaj način se vrši ažuriranje čvorova.

```
struct pokstablo *praznipreci (struct pokstablo *p) //rekurzivna funkcija koja
azurira broj praznih predaka u cvorovima i daje opt cvor kao rezultat
{
if (p==niz) p->prpreci=0;
else if (p->gore->prazan==1) p->prpreci=p->gore->prpreci+1;
    else p->prpreci=p->gore->prpreci;
if ((p->prpreci==opt->prpreci)&&(p->level>=opt->level)) opt=p; //uvek bira
dublji cvor sa istim brojem pr.predaka (tako dolazi do list cvora)
else if (p->prpreci>opt->prpreci) opt=p;
if (p->levo!=NULL) praznipreci(p->levo);
if (p->desno!=NULL) praznipreci(p->desno);
return(opt);
}
```

3.1.10.1 Rekurzivna funkcija „praznipreci“

Izbor list čvora vrši se proverom da li čvor na kome se nalazimo ima više praznih predaka od „opt“ čvora. Ukoliko ima on postaje novi „opt“ čvor. Takođe, ukoliko je broj praznih predaka čvora na kome se nalazimo jednak broju praznih predaka „opt“ čvora, a dubina čvora na kome se nalazimo veća ili jednaka od dubine „opt“ čvora, čvor na kome se nalazimo postaje novi „opt“ čvor. Na ovaj način se vrši kretanje na osnovu drugog kriterijuma ove tehnike, ali i vrši kretanje do list čvora. Zatim, funkcija se poziva za levi i za desni čvor ukoliko postoje. Na kraju, za izlaznu vrednost definiše se vrednost „opt“ pokazivača.

Premeštanje jednog list čvora u praznog pretka najmanje dubine dovodi do potrebe za ponovnim ažuriranjem vrednosti „prpreci“ čvorova stabla (tačnije svih potomaka čvora u koji je smešten list čvor). Na slici 3.1.10.2 prikazan je deo koda u kome se vrši ažuriranje i izbor list koda pomoću funkcije „praznipreci“.

```
opt=niz;
praznipreci(niz);
brpr=opt->prpreci; //potrebno za azuriranje stabla
tren=opt;
```

3.1.10.2 Deo koda u kome se vrši ažuriranje i izbor list čvora

Ovaj deo koda nalazi se na istom mestu u kodu kao i prethodno opisani delovi koda koji vrše izbor list čvora. Inicijalizacija pokzivača „opt“ je neophodna pre pozivanja funkcije „praznipreci“ radi pravilnog pronalaženja list čvora, jer kada je čvor „opt“ koren stabla, onda on ima najmanju vrednost praznih predaka. Funkcija „praznipreci“ poziva se sa pokazivačem na koren stabla, kako bi se izvršilo pravilno ažuriranje vrednosti „prpreci“ i izvršio izbor list čvora. Za

ispravno funkcionisanje dela koda koji vrši ažuriranje čvorova (predaka opt čvora) neophodno je ažurirati vrednost „brpr“, i podesiti pokazivač „tren“ na list čvor koji se izmešta.

3.1.11. Opis koda tehnike 6.2.

Tehnika 6.2. je tehnika u kojoj se izbor čvora vrši po kriterijumima: manji broj praznih predaka, veća dubina. Kao i kod tehnike 6.1, izbor list čvora vrši se prolaskom kroz sve čvorove stabla. Rekurzivna funkcija „praznipreci“ koja je izmenjena za potrebe ovog koda, prikazana je na slici 3.1.11.1. Kao i u kodu koji odgovara tehnici 6.1. ova funkcija vrši ažuriranje vrednosti „prpreci“ u svim čvorovima u stablu, ali vrši i izbor list čvora koji premeštamo. Ažuriranje vrednosti „prpreci“ vrši se na isti način kao i u kodu tehnike 6.1..

```
struct pokstablo *praznipreci (struct pokstablo *p) //rekurzivna funkcija koja
azurira broj praznih predaka u cvorovima i daje opt cvor kao rezultat
{
if (p==niz) p->prpreci=0;
else if (p->gore->prazan==1) p->prpreci=p->gore->prpreci+1;
    else p->prpreci=p->gore->prpreci;
if ((p->levo==NULL)&&(p->desno==NULL)&&(p->prpreci==opt->prpreci)&&(p-
>level>=opt->level)) opt=p; //uvek bira dublji cvor sa istim brojem pr.predaka
else if ((p->levo==NULL)&&(p->desno==NULL)&&(p->prpreci<opt->prpreci)&&(p-
>prpreci>0)) opt=p;
if (p->levo!=NULL) praznipreci(p->levo);
if (p->desno!=NULL) praznipreci(p->desno);
return(opt);
}
```

Slika 3.1.11.1 Rekurzivna funkcija „praznipreci“

Jedina izmena u odnosu na funkciju „praznipreci“ iz koda tehnike 6.1. tiče se izbora list čvora. Izbor čvora se vrši na sledeći način. Ukoliko je čvor na kome se nalazimo list čvor, čija je vrednost broja praznih predaka manja od vrednosti praznih predaka kod „opt“ čvora, a veća od nule, „opt“ čvor postaje čvor na kome se nalazimo. Takođe, ukoliko je čvor na kome se nalazimo list čvor, čija je vrednost broja praznih predaka jednaka vrednosti praznih predaka kod „opt“ čvora, a dubina veća ili jednaka od dubine „opt“ čvora, „opt“ čvor postaje čvor na kome se nalazimo. Ovime se se vrši izbor na osnovu drugog kriterijuma – veće dubine. Nakon toga, funkcija se poziva za levi i za desni čvor ukoliko postoje. Na kraju, za izlaznu vrednost definiše se vrednost „opt“ pokazivača.

Kako je primarni kriterijum ove tehnike manji broj praznih predaka, inicijalizacija promenljivih pre pokretanja funkcije „praznipreci“ razlikuje se u odnosu na način na koji je to učinjeno kod tehnike 6.1.. Vrednost pokazivača „opt“ se ne inicijalizuje dodeljivanjem vrednosti pokazivača na koren stabla, već na vrednost pokazivača „opt1“. Pokazivač „opt1“ se inicijalizuje pre dela koda koji vrši transformaciju binarnog stabla u stablo sa prioritetima (slika 3.1.10.2). Najpre se vrši alociranje memorije za čvor „opt1“, a zatim se tom čvoru dodeljuje maksimalna vrednost broja praznih predaka.

```
opt1=malloc(sizeof(stablo));
opt1->prpreci=32;
```

Slika 4.1.10.2 Inicijalizacija pokazivača „opt1“

Deo koda u kome se vrši izbor čvora prikazan je na slici 3.1.10.3. Jedina razlika u odnosu na kod tehnike 6.1. jeste u načinu inicijalizacije pokazivača „opt“ pre pozivanja funkcije „praznipreci“.

Pokazivaču „opt“ dodeljuje se vrednost pokazivača „opt1“ jer taj čvor ima najveću vrednost broja praznih predaka. Pozivanje funkcije „praznipreci“, ažuriranje vrednosti „brpr“ i pokazivača „tren“ vrši se na isti način kao i u kodu tehnike 6.1..

```
opt=opt1;
praznipreci(niz);
brpr=opt->prpreci; //potrebno za azuriranje stabla
tren=opt;
```

Slika 4.1.10.3 Deo koda u kome se vrši ažuriranje i izbor list čvora

4. ANALIZA PERFORMANSI STABLA SA PRIORITETIMA

U ovom delu rada biće izvršena analiza tehnika formiranja stabla opisanih u trećem poglavlju ovog rada. Takođe, analizom ćemo utvrditi u kojoj meri je formiranje stabla sa prioritetima opravdano. Stabla sa prioritetima su kreirana na osnovu RIB (*Routing Information Base*) tabela. RIB tabela je tabela koja se nalazi u ruterima i sadrži najmanje tri vrste podataka: IP prefiks, metrika (cena) i sledeći hop (*gateway* adresa, izlazni port rutera). Na osnovu RIB tabele ruter je u stanju da pristigli IP paket prosledi na pravu destinaciju. Za potrebe analize tehnika formiranja stabla sa prioritetima korišćene su RIB tabele preuzete sa veb stranice: http://www.ripe.net/data-tools/stats/ris/ris-raw-data[5]. Stabla sa prioritetima kreirana su pomoću kodova opisanih u poglavlju 3 ovog rada. Kodovi su priloženi uz tezu na CD-u.

Tehnike formiranja stabla sa prioritetima opisane u poglavlju 3, su tehnike transformacije binarnog stabla u stablo sa prioritetima. Deo koda koji vrši učitavanje RIB tabela i formiranje binarnog stabla preuzet je iz rada [4]. Analiza tehnika formiranja stabala sa prioritetima biće vršena na osnovu parametara dobijenih pokretanjem funkcije „statistika“ čiji je kod opisan u poglavlju 3 ovog rada.

U nastavku će biti navedeni parametri na osnovu kojih ćemo vršiti analizu tehnika za formiranje stabla sa prioritetima. Dubina stabla je parametar koji odgovara maksimalnoj vrednosti nivoa koju (barem) jedan čvor u tom stablu ima. Prema tome, za IPv4 adrese, maksimalna vrednost dubine stabla je 32. Takođe, posmatraćemo i vrednosti dubine levog i desnog podstabla korena stabla. Broj čvorova u stablu zavisi od broja prefiksa u slučaju stabala sa prioritetima, ali i od vrednosti prefiksa u slučaju binarnih stabala. Broj popunjenih čvorova jednak je broju prefiksa u RIB tabeli. Broj praznih čvorova binarnog stabla predstavlja razliku broja čvorova i broja popunjenih čvorova. Rezultati analize veoma zavise od broja praznih čvorova u stablu i njihovog rasporeda, jer se u njih smeštaju list čvorovi. Broj čvorova sa simetričnom dubinom predstavlja broj čvorova čije levo i desno podstablo imaju istu dubinu. Broj čvorova sa simetričnom decom predstavlja broj čvorova koji imaju dete i sa leve i sa desne strane, odnosno imaju oba aktivna pokazivača. Broj čvorova sa asimetričnom decom predstavlja broj čvorova koji ima samo jedan aktivan pokazivač. Broj listova predstavlja broj list čvorova, odnosno broj čvorova koji nemaju decu. Broj čvorova sa simetričnim listovima predstavlja broj čvorova koji ima list čvor i sa leve i sa desne strane. Broj čvorova sa asimetričnim listovima predstavlja broj čvorova koji imaju jedan list čvor sa jedne strane, a sa druge nemaju list čvor (sa druge strane ili nemaju decu ili imaju više od jednog deteta). Broj čvorova sa jednim listom predstavlja broj čvorova koji imaju samo jedan aktivan pokazivač koji pokazuje na list čvor. Broj optimizacija jednak je vrednosti broja izmeštenih list čvorova. Za analizu se koriste i vrednosti brojeva čvorova i broja popunjenih čvorova po nivoima. Trajanje formiranja stabla sa prioritetima predstavlja vremenski interval potreban za formiranje stabla sa prioritetima na osnovu RIB tabele, i direktno zavisi od performansi računara na kome se pokreću programi kreirani na osnovu od opisanih kodova. Za potrebe ovog rada, pre i tokom testiranja nisu vršena dodatna podešavanja softvera (podešavanja operativnog sistema i podešavanja Code::blocks softverskog alata). Karakteristike računara na kome su pokretani programi za potrebe ovog rada su:

[1] Chipset: Intel G33 Express

[2] CPU: Intel Core 2 Quad Q9300 Processor

[3] RAM: 8-GB DDR2 Synch DRAM PC2-6400

[4] OS: Windows 7 Professional (64bit version)

Analiza tehnika formiranja stabla sa prioritetima rađena je na osnovu 8 RIB tabela preuzetih sa veb stranice http://www.ripe.net/data-tools/stats/ris/ris-raw-data[5]. U obzir su uzete dve veličine RIB tabela: tabele sa oko 250 hiljada prefiksa i tabele sa oko 500 hiljada prefiksa. Za obe veličine RIB tabela, analiza je rađena za tabele sa dva različita rutera. Za obe veličine RIB tabele i za svaki ruter, analiza je rađena za dve RIB tabele formirane u dva bliska meseca (dva meseca vremenske razlike). U tabeli 4.1 prikazan je spisak RIB tabela korišćenih za analizu tehnika formiranja stabla sa prioritetima.

Tabela 4.1 RIB tabele korišćene za analizu

	Ruter	**Datum formiranja RIB tabele**	**Broj prefiksa**
RIB.1	RRC05 Vienna	15.03.2008	249,719
RIB.2	RRC05 Vienna	15.05.2008	256,264
RIB.3	RRC12 Frankfurt	01.06.2008	260,630
RIB.4	RRC12 Frankfurt	01.08.2008	266,659
RIB.5	RRC05 Vienna	15.05.2014	494,420
RIB.6	RRC05 Vienna	15.07.2014	502,605
RIB.7	RRC10 Milan	15.05.2014	495,385
RIB.8	RRC10 Milan	15.07.2014	503,158

Navedene RIB tabele i kompletni rezultati svih kodova za navedene RIB tabele priloženi su na CD-u zajedno sa kodovima i tezom. Prvo ćemo vršiti analizu rezultata tehnika formiranja stabla sa prioritetima za RIB tabele koje sadrže oko 250 hiljada prefiksa, a zatim za RIB tabele koje sadrže oko 500 hiljada prefiksa.

U tabeli 4.2 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.1 sa rutera RRC05 Vienna. Ova RIB tabela sadrži 249719 prefiksa. Prvi red u tabeli odnosi se na binarno stablo, odnosno na stablo pre transformacije. Preostali redovi odnose se na tehnike transformacije binarnog stabla u stablo sa prioritetima. U tabeli 4.3 prikazan je broj čvorova po nivoima za tabelu RIB.1 sa rutera RRC05 Vienna. Prvi red u tabeli odnosi se na ukupan broj čvorova po nivoima za binarno stablo. Drugi red se odnosi na broj popunjenih čvorova po nivoima za binarno stablo. Ostali redovi sadrže vrednosti broja (popunjenih) čvorova po nivoima za stablo sa prioritetima formiranog pomoću svih tehnika predstavljenih u poglavlju 3.

Tabela 4.2 Rezultati formiranja stabla sa prioritetima za RIB.1 tabelu

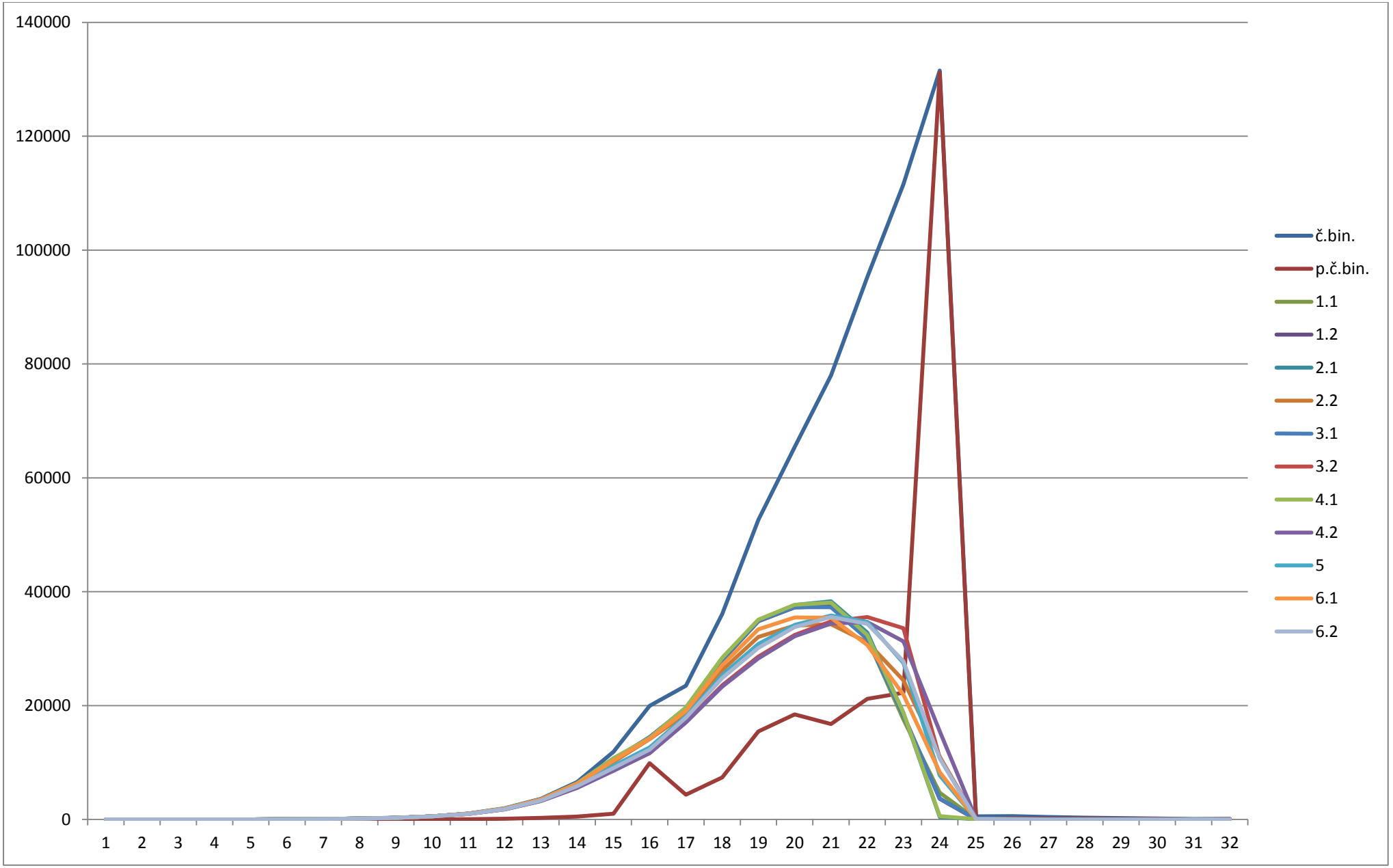
	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	642558	233545	409012	392838	249719	95632	154087	395049	227725	187107	227726	64325	99076	68501	/	/
1.1	26	26	26	249719	95632	154086	0	249719	95632	154086	127779	93588	62542	93589	11783	70023	45129	195825	38.034
1.2	26	26	26	249719	95632	154086	0	249719	95632	154086	133014	95919	57880	95920	14313	67294	42248	197370	37.697
2.1	26	26	25	249719	95632	154086	0	249719	95632	154086	150106	100487	48744	100488	19784	60920	38473	203287	37.701
2.2	27	27	26	249719	95631	154087	0	249719	95631	154087	117906	89551	70616	89552	7515	74522	44775	192037	36.895
3.1	26	26	26	249719	95632	154086	0	249719	95632	154086	132771	95838	58042	95839	14238	67363	42303	197222	37.206
3.2	30	30	30	249719	95631	154087	0	249719	95631	154087	104505	78838	92042	78839	6807	65225	43605	214803	37.092
4.1	26	26	26	249719	95632	154086	0	249719	95632	154086	148894	100350	49018	100351	19606	61139	38513	202970	36.969
4.2	30	30	30	249719	95631	154087	0	249719	95631	154087	95702	74376	100966	74377	1580	71217	46803	207695	36.999
5	29	29	29	249719	95631	154087	0	249719	95631	154087	114253	85347	79024	85348	9643	66062	43389	208860	37.592
6.1	26	26	26	249719	95632	154086	0	249719	95632	154086	123833	91912	65894	91913	10471	70971	43384	193614	2697.45
6.2	30	30	29	249719	95631	154087	0	249719	95631	154087	104821	79284	91150	79285	4801	69683	47832	214580	3235.34

Tabela 4.3 Broj čvorova po nivoima za RIB.1 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	26	48	91	166	303	563	1035	1953	3623	6598	11883	19941	23470	36080	52613	65414	77957	95241	111549	131517	563	577	421	325	241	166	80	86
p.č.bin.	1	1	0	0	2	3	10	27	9	16	40	140	283	505	1003	9890	4344	7379	15456	18450	16769	21197	22258	131129	83	200	128	123	94	93	0	86
1.1	2	4	7	14	26	48	90	164	291	545	1007	1895	3507	6317	10429	14455	19491	27898	34834	37267	37373	31752	17555	4712	29	6	0	0	0	0	0	0
1.2	2	4	7	14	26	48	90	164	291	545	1007	1894	3507	6326	10474	14411	19512	27974	34836	37230	37487	31856	18361	3617	27	8	0	0	0	0	0	0
2.1	2	4	7	14	26	48	90	164	291	545	1008	1897	3506	6331	10727	14229	19577	28301	35004	37665	38320	32821	18683	427	30	1	0	0	0	0	0	0
2.2	2	4	7	14	26	48	89	163	287	541	1005	1876	3475	6188	10095	14100	18539	26232	32069	34040	34269	31118	24449	11020	38	22	2	0	0	0	0	0
3.1	2	4	7	14	26	48	90	164	292	545	1008	1897	3507	6331	10485	14471	19583	28026	34908	37269	37279	31693	18316	3712	33	8	0	0	0	0	0	0
3.2	2	4	7	14	26	47	87	153	285	528	977	1795	3232	5614	8672	11656	17103	23560	28593	32413	34797	35561	33596	10793	114	43	23	11	7	5	0	0
4.1	2	4	7	14	26	48	90	164	292	545	1008	1899	3507	6343	10746	14309	19671	28399	35106	37685	38082	32390	18751	591	34	5	0	0	0	0	0	0
4.2	2	4	7	14	26	47	87	153	285	528	975	1790	3222	5560	8534	11635	17020	23347	28276	32153	34382	34672	31293	15489	116	53	23	11	9	5	0	0
5	2	4	7	14	26	47	88	158	286	535	994	1842	3361	5955	9410	12665	18108	25370	30820	34161	35805	34687	27489	7746	71	40	16	6	5	0	0	0
6.1	2	4	7	14	26	48	89	164	288	541	1005	1883	3489	6253	10326	14205	19059	27093	33409	35459	35434	30647	21922	8301	35	15	0	0	0	0	0	0
6.2	2	4	7	14	26	47	90	162	289	534	987	1824	3328	5864	9031	12261	17886	24821	30147	33795	35562	34358	27796	10720	85	41	21	9	6	1	0	0

Jedan od ciljeva transformacije binarnog stabla u stablo sa prioritetima jeste uprošćavanje stabla i smanjivanje njegove dubine. U tabeli 4.2 možemo videti da su tehnike 1.1, 1.2, 2.1, 3.1, 4.1 i 6.1 postigle najbolje rezultate u pogledu dubine stabla. Međutim, stablo manje dubine nije nužno i bolje stablo. Veoma bitna stvar je raspodela čvorova, odnosno prefiksa u stablu. Od interesa je smestiti što više prefiksa u niže nivoe stabla, kako bi se lukap funkcija za što više IP paketa brže okončala. Prema tome, najbolja tehnika je ona koja formira stablo što manje dubine sa što više čvorova u nižim nivoima, bliže korenu stabla. Ovakvo stablo ima i najveću simetriju. Međutim, rezultati najviše zavise od same RIB tabele, odnosno od binarnog stabla formiranog na osnovu nje. Pošto tehnike transformacije binarnog stabla vrše premeštanje list čvora u prazan čvor najniže dubine, rezultati u velikoj meri zavise od broja i raspodele praznih čvorova u stablu. Stepen simetrije stabla možemo utvrditi na osnovu sledećih vrednosti: broj čvorova sa simetričnom dubinom, broj čvorova sa simetričnom decom, broj čvorova sa asimetričnom decom, broj čvorova sa simetričnim listovima, broj čvorova sa asimetričnim listovima i broj čvorova sa jednim listom. Na osnovu ovih vrednosti u tabeli 4.2, možemo zaključiti da najveću simetriju imaju stabla sa prioritetima formirana pomoću tehnika 2.1 i 4.1, a najmanju simetriju ima stablo formirano pomoću tehnike 3.2. Tehnika koja izvrši transformaciju binarnog stabla nakon manjeg broja optimizacija, predstavlja bolje rešenje. Na osnovu vrednosti broja optimizacija u tabeli 4.2, možemo zaključiti da su tehnike 2.2 i 6.1 tehnike koje vrše transformaciju sa najmanje optimizacija. Međutim, razlike u broju optimizacija između tehnika se ne razlikuju značajno (razlika reda 5%). Takođe, kod različitih tehnika, optimizacija podrazumeva različit broj upita, i različit broj pristupa memoriji. Zbog toga posmatramo i vrednost trajanja formiranja stabla sa prioritetima. Na osnovu vrednosti u tabeli 4.2, možemo zaključiti da tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje. Ove tehnike, nakon svakog premeštanja list čvora moraju izvršiti ažuriranje vrednosti praznih predaka svih čvorova u stablu. Ovo zahteva veliki broj pristupa memoriji što i predstavlja najsporiji proces. Nakon izmeštanja list čvora, potrebno je ažurirati sve potomke čvora u koji je smešten list čvor. Međutim, pošto funkcija koja vrši ažuriranje ujedno i vrši izbor list čvora, pokretanje funkcije sa poslednjim list čvorom (kao ulaznim parametrom) ne predstavlja rešenje. Razlika u trajanju formiranja stabla ostalih tehnika je zanemarljivo mala (razlika reda 1%).

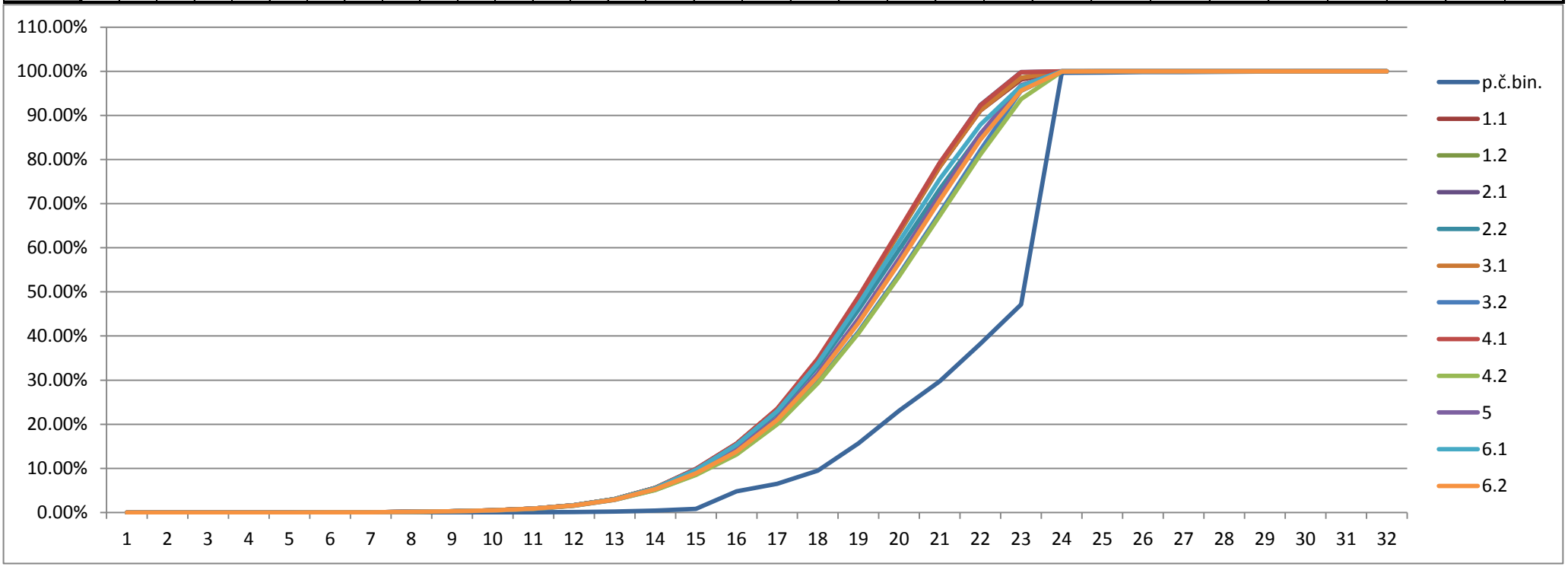
U tabeli 4.3 možemo videti da stabla sa prioritetima formirana na osnovu tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Kod ovih stabala možemo uočiti veći broj čvorova u nižim nivoima, kao i da je manji broj čvorova u višim nivoima u odnosu na ostale tehnike. Tako, na nivou 24 broj čvorova kod ovih stabala smanjen je sa 131 hiljade na svega 427 čvorova kod tehnike 2.1, odnosno na 591 čvorova kod tehnike 4.1. Tehnike formiranja stabla 1.2 i 3.1 predstavljaju dobro rešenje u pogledu raspodele čvorova po nivoima, ali ne vrše smanjivanje broja čvorova na nivou 24 u meri u kojoj to postižu tehnike 2.1 i 4.1. Lošu raspodelu čvorova po nivoima možemo uočiti kod stabala kreiranih pomoću tehnika 2.2 i 3.2. Na slici 4.1 dat je grafički prikaz tabele 4.3. Ako uporedimo liniju koja predstavlja broj čvorova binarnog stabla sa linijama koje predstavljaju broj čvorova stabala sa prioritetima, vidimo u kojoj meri dolazi do uprošćavanja stabla i uštede memorijskih resursa. Ako uporedimo liniju koja predstavlja broj popunjenih čvorova kod binarnog stabla sa linijama koje predstavljaju brojeve (popunjenih) čvorova stabala sa prioritetima, možemo videti raspodelu prefiksa po nivoima, samim tim, možemo videti na kojim nivoima se mogu doneti odluke o prosleđivanju za najviše prefiksa. Takođe možemo primetiti izražene skokove kod linija koje predstavljaju binarno stablo, i to na nivoima 16 i 24. Skokovi su posledica nekadašnje upotrebe klasnog adresiranja. Kod linija koje predstavljaju stabla sa prioritetima, ovi skokovi ne postoje. Odluka o prosleđivanju paketa čiji je odgovarajući prefiks dužine 24, kod ovih stabala u većini slučajeva donosiće se pre dolaska do 24-tog nivoa u stablu. Tabela 4.4 formirana je na osnovu tabele 4.3 i prikazuje odnos kumulativnog broja popunjenih čvorova po nivoima i ukupnog broja čvorova. Drugim rečima, tabela prikazuje procenat popunjenih čvorova smeštenih do određenog



Slika 4.1 Grafik raspodele čvorova po nivoima za tabelu RIB.1.

Tabela 4.4 Odnos kumulativnog broja popunjenih čvorova po nivoima i ukupnog broja čvorova za tabelu RIB.1 [%]

nivo:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
p.č.bin.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.4	0.8	4.8	6.5	9.5	15.7	23.0	29.8	38.3	47.2	99.7	99.7	99.8	99.8	99.9	99.9	100.0	100.0	100.0
1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.6	9.7	15.5	23.3	34.5	48.5	63.4	78.4	91.1	98.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.6	9.8	15.5	23.4	34.6	48.5	63.4	78.4	91.2	98.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.6	9.9	15.6	23.4	34.7	48.8	63.8	79.2	92.3	99.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.5	9.5	15.2	22.6	33.1	46.0	59.6	73.3	85.8	95.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.6	9.8	15.6	23.4	34.6	48.6	63.5	78.5	91.2	98.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	2.9	5.1	8.6	13.3	20.1	29.5	41.0	54.0	67.9	82.1	95.6	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.6	9.9	15.6	23.5	34.9	48.9	64.0	79.3	92.2	99.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	2.9	5.1	8.5	13.2	20.0	29.3	40.7	53.5	67.3	81.2	93.7	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	2.9	5.3	9.1	14.2	21.4	31.6	43.9	57.6	71.9	85.8	96.8	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
6.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	3.0	5.5	9.7	15.4	23.0	33.8	47.2	61.4	75.6	87.9	96.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
6.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.3	0.5	0.9	1.6	2.9	5.3	8.9	13.8	21.0	30.9	43.0	56.5	70.8	84.5	95.6	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0



Slika 4.2 Grafički prikaz odnosa kumulativnog broja popunjenih čvorova po nivoima i ukupnog broja čvorova za tabelu RIB.1 [%]

nivoa. Na osnovu tabele 4.4 grubo možemo zaključiti u kojoj meri se na određenim nivoima lukap funkcija okončava. Takođe, možemo uočiti da je 3 puta više prefiksa smešteno u delu stabla od korena stabla do nivoa 16 kod stabala sa prioritetima nego kod binarnog stabla. Takođe, kod stabala sa prioritetima, oko 40% više prefiksa smešteno je u delu stabla ispod nivoa 20 nego što je to slučaj kod binarnog stabla. Tabela 4.4 grafički je prikazana na slici 4.2. Na grafiku jasno možemo videti da stablo sa prioritetima u značajnoj meri može povećati performanse lukap funkcije. Kod stabala sa prioritetima vidimo da se oko 50% prefiksa nalazi se između korena stabla i nivoa 19, dok kod binarnog stabla, oko 50% prefiksa se nalazi između korena stabla i nivoa 23. Takođe, možemo uočiti da se oko 50% ukupnih popunjenih čvorova kod binarnog stabla nalazi na nivou 24, dok se kod stabala formiranih na osnovu tehnika 2.1 i 4.1 na nivou 24 nalazi svega 0.2% odnostno 0.3% ukupnih popunjenih čvorova.

Na osnovu tabela 4.2, 4.3 i 4.4 možemo zaključiti da za RIB.1 tabelu, stabla sa prioritetima formirana pomoću svih navedenih tehnika dovešće do povećanja performansi lukap funkcije u odnosu na binarno stablo. Stabla formirana pomoću tehnika 4.1 i 2.1 predstavljaju najbolje rešenje zbog velike simetrije i povoljne raspodele čvorova u stablu (bliže korenu stabla). Tehnike 3.1 i 1.2 predstavljaju dobra rešenja. Ako uporedimo tehnike 1.1 i 1.2, možemo zaključiti da tehnika 1.2 vrši formiranje stabla sa većom simetrijom. To je posledica činjenice da tehnika 1.2 za treći kriterijum ima metodu slučajnog izbora. Takođe možemo uočiti da tehnika 4.2 formira stablo sa najnepovoljnijom raspodelom čvorova, odnosno, veliki broj čvorova nalazi se u višim nivoima stabla. Tehnika 5 ne predstavlja dobro rešenje jer ne daje dobru raspodelu čvorova po nivoima i suprotno očekivanju, ne formira stablo sa velikom simetrijom. Tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje, pre svega zbog znatno kompleksnijeg i dugotrajnijeg procesa transformacije binarnog stabla.

U tabeli 4.5 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.2 sa rutera RRC05 Vienna. Ova RIB tabela sadrži 256264 prefiksa. Stabla koja imaju najmanju dubinu (26) formirana su pomoću tehnika 1.1, 1.2, 2.1, 3.1, 4.1 i 6.1. Stabla formirana pomoću tehnika 3.2, 4.2 i 6.2 imaju najveću dubinu (30). Stabla sa prioritetima koja imaju najveću simetriju su stabla kreirana pomoću tehnika 2.1 i 4.1. Ova stabla imaju najveći broj čvorova sa podstablima iste dubine, čvorova sa oba aktivna pokazivača i čvorova sa dva deteta koja su list čvorovi. Stablo koje ima najmanju simetriju je formirano tehnikom 3.2. Stabla koja formirana pomoću tehnika 4.2, 5 i 6.2 takođe ne predstavljaju dobro rešenje zbog male simetrije stabla. Tehnike 2.2 i 6.1 vrše transformaciju binarnog stabla u stablo sa prioritetima sa najmanjim brojem optimizacija, dok je najviše optimizacija potrebno tehnikama 3.2 i 6.2. Što se tiče vremena potrebnog za formiranje stabla sa prioritetima, možemo uočiti da sve tehnike formiranja stabla sa prioritetima osim tehnika 6.1 i 6.2 vrše transformaciju za približno isto vreme. Tehnike 6.1 i 6.2 u ovom pogledu, ne predstavljaju dobro rešenje.

U tabeli 4.6 prikazan je broj čvorova po nivoima za tabelu RIB.2 sa rutera RRC05 Vienna. Na osnovu tabele 4.6 možemo zaključiti da stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Broj čvorova binarnog stabla najveći je na nivou 24 i iznosi oko 135 hiljada. Broj čvorova na ovom nivou smanjen je na 452 čvora tehnikom 2.1 odnosno na 617 čvorova tehnikom 4.1. U pogledu broja čvorova po nivoima, tehnike 1.1, 1.2, 3.1 i 6.1 imaju nešto manje povoljnu raspodelu čvorova. Stablo formirano pomoću tehnika 3.2, 5 i 6.2 nema povoljnu raspodelu, dok tehnika 4.2 formira stablo sa najnepovoljnijom raspodelom čvorova.

Tabela 4.5 Rezultati formiranja stabla sa prioritetima za RIB.2 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	656971	244026	412944	400706	256264	100441	155823	405694	233630	189710	233631	66410	100811	69584	/	/
1.1	26	26	26	256264	100441	155822	0	256264	100441	155822	131108	96016	64231	96017	12076	71865	46373	200633	38.418
1.2	26	26	26	256264	100441	155822	0	256264	100441	155822	136584	98396	59471	98397	14714	68969	43382	202174	38.253
2.1	26	26	26	256264	100441	155822	0	256264	100441	155822	154183	103136	49991	103137	20373	62391	39457	208335	38.291
2.2	28	28	26	256264	100440	155823	0	256264	100440	155823	121124	91935	72393	91936	7799	76338	45871	196690	38.316
3.1	26	26	26	256264	100441	155822	0	256264	100441	155822	136225	98302	59659	98303	14517	69269	43590	202094	38.161
3.2	30	30	30	256264	100440	155823	0	256264	100440	155823	107614	81102	94059	81103	7105	66893	44571	220141	38.244
4.1	26	26	26	256264	100441	155822	0	256264	100441	155822	152805	102955	50353	102956	20090	62776	39551	207977	38.004
4.2	30	30	30	256264	100440	155823	0	256264	100440	155823	98569	76483	103297	76484	1646	73192	47976	212870	38.002
5	29	29	29	256264	100441	155822	0	256264	100441	155822	117732	87756	80751	87757	10122	67513	44345	214045	38.165
6.1	26	26	26	256264	100441	155822	0	256264	100441	155822	127047	94261	67741	94262	10733	72796	44560	198322	2821.92
6.2	30	30	29	256264	100440	155823	0	256264	100440	155823	107667	81327	93609	81328	4903	71522	49189	219924	3381.39

Tabela 4.6 Broj čvorova po nivoima za RIB.2 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	26	49	91	164	303	565	1046	1976	3672	6682	12029	20211	23783	36471	53399	66671	79748	97523	114526	135355	589	602	452	350	275	188	95	102
p.č.bin.	1	1	0	0	2	3	10	26	9	16	41	141	286	514	1010	10026	4431	7397	15622	18868	17409	21780	22875	134945	85	202	133	119	109	101	0	102
1.1	2	4	7	14	26	49	90	163	296	551	1024	1922	3554	6393	10563	14674	19709	28343	35572	38328	38644	33130	18246	4916	33	10	0	0	0	0	0	0
1.2	2	4	7	14	26	49	90	163	296	551	1025	1921	3554	6397	10616	14627	19722	28401	35591	38325	38733	33220	19038	3851	33	7	0	0	0	0	0	0
2.1	2	4	7	14	26	49	90	163	296	551	1025	1923	3553	6405	10849	14454	19799	28753	35767	38762	39566	34275	19440	452	35	3	0	0	0	0	0	0
2.2	2	4	7	14	26	49	90	162	292	547	1019	1906	3522	6266	10234	14292	18767	26642	32856	34937	35407	32297	25398	11460	42	21	3	1	0	0	0	0
3.1	2	4	7	14	26	49	90	163	297	551	1025	1922	3556	6401	10631	14680	19792	28432	35672	38386	38552	33120	19003	3844	33	11	0	0	0	0	0	0
3.2	2	4	7	14	26	48	88	154	288	537	990	1816	3271	5689	8769	11785	17347	23917	29194	33197	35882	36960	34860	11207	113	48	23	14	9	4	0	0
4.1	2	4	7	14	26	49	90	163	297	551	1025	1925	3555	6417	10874	14530	19899	28823	35894	38768	39311	33853	19527	617	38	4	0	0	0	0	0	0
4.2	2	4	7	14	26	48	88	154	288	537	989	1808	3265	5643	8632	11763	17274	23687	28851	32978	35497	35970	32476	16036	117	56	24	14	10	5	0	0
5	2	4	7	14	26	48	89	159	289	540	1009	1870	3406	6052	9530	12818	18380	25761	31473	34963	37004	36095	28616	7950	87	38	16	11	6	0	0	0
6.1	2	4	7	14	26	49	90	163	293	547	1020	1911	3538	6330	10469	14396	19289	27512	34143	36492	36599	31864	22796	8654	37	18	0	0	0	0	0	0
6.2	2	4	7	14	26	48	90	161	295	540	1003	1845	3369	5934	9148	12416	18132	25178	30775	34618	36708	35713	28885	11180	87	45	21	11	7	1	0	0

Analizom rezultata kreiranih za RIB.2 tabelu, možemo doći do zaključka da tehnike formiranja stabla sa prioritetima 2.1 i 4.1 predstavljaju najbolje rešenje. Ove tehnike formiraju stabla koja imaju najmanju dubinu, najveću simetriju i najbolju raspodelu čvorova po nivoima. Nešto lošije rezultate imaju tehnike 3.1 i 1.2. Kao i u prethodnom slučaju možemo uočiti manji nivo simetrije kod stabla formiranog tehnikom 1.1 u odnosu na ono formirano tehnikom 1.2. Tehnike 2.2, 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer vrše formiranje stabla sa nepovoljnom raspodelom čvorova i niskim nivoom simetrije stabla. Takođe, tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje pre svega zbog trajanja formiranja stabla.

Sada ćemo analizirati rezultate kreirane za RIB.3 tabelu sa rutera RRC12 Frankfurt. Ova RIB tabela sadrži 260630 prefiksa. U tabeli 4.7 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.3. Analizom tabele, možemo utvrditi da stabla formirana pomoću tehnika 1.1, 1.2, 2.1, 3.1 i 4.1 imaju najmanje dubine (dubina 29). Kod tehnika 2.1 i 4.1 levo podstablo (u odnosu na koren stabla) ima dubinu 28. Stablo formirano pomoću tehnike 4.2 ima najveću dubinu koja iznosi 32, dok dubina stabala kreiranih pomoću tehnika 6.2 i 3.2 iznosi 31. Najveći stepen simetrije imaju stabla formirana pomoću tehnika 2.1 i 4.1. Najmanji stepen simetrije imaju stabla formirana pomoću tehnika 4.2, 3.2 i 6.2. Transformacija stabla obavlja se sa najmanjim brojem optimizacija pomoću tehnika 6.1 i 2.2, dok je najveći broj optimizacija potreban tehnikama 3.2 i 6.2. Što se tiče vremena potrebnog za transformaciju, možemo uočiti da sve tehnike formiranja stabla sa prioritetima osim tehnika 6.1 i 6.2 vrše transformaciju za približno isto vreme. Tehnike 6.1 i 6.2 u ovom pogledu, ne predstavljaju dobro rešenje.

U tabeli 4.8 prikazan je broj čvorova po nivoima za tabelu RIB.3 sa rutera RRC12 Frankfurt. Na osnovu tabele 4.8 možemo zaključiti da stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Broj čvorova binarnog stabla najveći je na nivou 24 i iznosi oko 138 hiljada. Broj čvorova na ovom nivou smanjen je na 742 čvora tehnikom 2.1 odnosno na 897 čvorova tehnikom 4.1. U pogledu broja čvorova po nivoima, tehnike 1.1, 1.2, 3.1 i 6.1 imaju nešto manje povoljnu raspodelu čvorova, što je najizraženije u višim nivoima (veći broj čvorova). Stabla formirana pomoću tehnika 5 i 6.2 nemaju povoljnu raspodelu čvorova po nivoima, dok tehnike 3.2 i 4.2 formiraju stabla sa najnepovoljnijom raspodelom čvorova. Na slici 4.3 dat je grafički prikaz raspodele čvorova po nivoima. Na slici 4.3 jasno se vidi raspodela čvorova po nivoima, kao i efekat transformacija binarnog stabla u stabla sa prioritetima. Takođe, možemo uočiti da linija koja predstavlja broj popunjenih čvorova po nivoima za binarno stablo ima nagle skokove na nivou 16 i 24 zbog klasnog adresiranja. Ovaj efekat poptuno je potisnut kod svih stabala sa prioritetima.

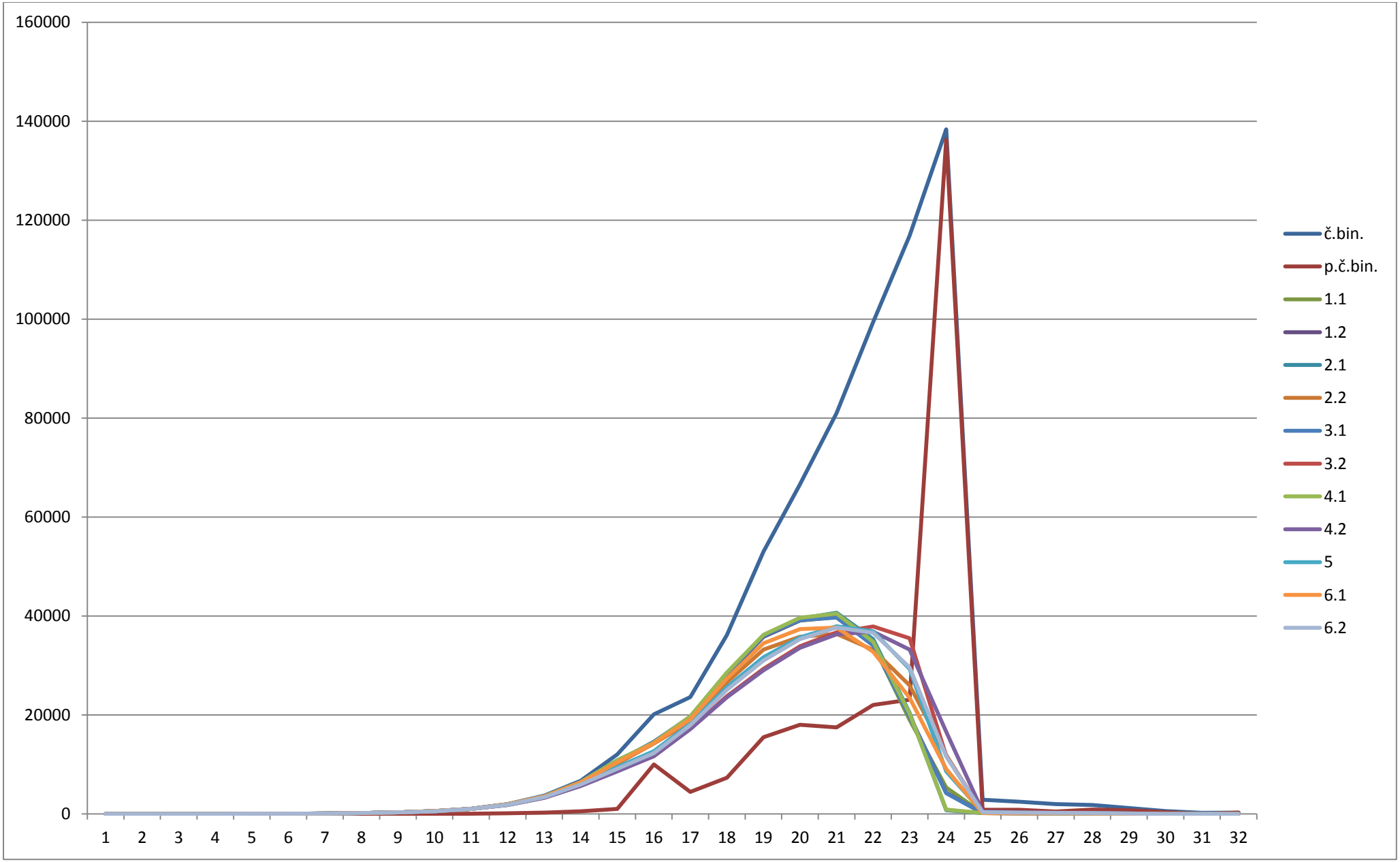
Analizom rezultata kreiranih za RIB.3 tabelu, možemo doći do zaključka da tehnike formiranja stabla sa prioritetima 2.1 i 4.1 predstavljaju najbolje rešenje. Nešto lošije rezultate imaju tehnike 3.1 i 1.2. Tehnike 2.2, 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer vrše formiranje stabla sa nepovoljnom raspodelom čvorova i niskim nivoom simetrije stabla. Takođe, tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje pre svega zbog trajanja formiranja stabla.

Tabela 4.7 Rezultati formiranja stabla sa prioritetima za RIB.3 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	673262	251878	421383	412631	260630	103054	157576	408670	237581	198099	237582	67203	103176	71303	/	/
1.1	29	29	29	260630	103054	157575	0	260630	103054	157575	133211	97605	65419	97606	12238	73130	47117	204236	76.565
1.2	29	29	29	260630	103054	157575	0	260630	103054	157575	138417	99928	60773	99929	14808	70313	44153	205775	76.552
2.1	29	28	29	260630	103054	157575	0	260630	103054	157575	156145	104723	51183	104724	20502	63720	40219	212008	75.311
2.2	30	30	30	260630	103053	157576	0	260630	103053	157576	123009	93418	73793	93419	7940	77539	46635	200178	76.693
3.1	29	29	29	260630	103054	157575	0	260630	103054	157575	138757	100104	60421	100105	14986	70133	43981	205787	75.885
3.2	31	31	31	260630	103053	157576	0	260630	103053	157576	108604	82180	96269	82181	7091	67999	45468	224023	75.104
4.1	29	28	29	260630	103054	157575	0	260630	103054	157575	154776	104548	51533	104549	20204	64141	40352	211597	75.672
4.2	32	32	31	260630	103053	157576	0	260630	103053	157576	99466	77556	105517	77557	1720	74117	48753	216697	75.641
5	30	30	30	260630	103054	157575	0	260630	103054	157575	118894	89053	82523	89054	10117	68820	45115	217720	75.848
6.1	30	30	30	260630	103054	157575	0	260630	103054	157575	129165	95864	68901	95865	10883	74099	45330	201851	3042.82
6.2	31	31	30	260630	103053	157576	0	260630	103053	157576	109128	82641	95347	82642	4960	72722	49925	223809	3545.68

Tabela 4.8 Broj čvorova po nivoima za RIB.3 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	26	49	91	165	304	568	1052	1987	3687	6691	12032	20148	23615	36141	53030	66593	81020	99409	116859	138362	2880	2473	1967	1811	1206	566	231	271
p.č.bin.	0	0	0	0	0	0	0	19	9	16	43	143	285	518	1026	9977	4440	7332	15510	18030	17504	22017	23077	136260	840	823	479	885	754	372	0	271
1.1	2	4	7	14	26	49	90	164	298	556	1031	1929	3561	6387	10526	14560	19521	28134	35808	39079	39788	34165	19028	5347	172	153	121	83	26	0	0	0
1.2	2	4	7	14	26	49	90	164	298	556	1031	1929	3561	6391	10578	14521	19530	28180	35827	39097	39823	34235	19882	4280	176	151	121	81	25	0	0	0
2.1	2	4	7	14	26	49	90	164	298	556	1031	1931	3561	6404	10814	14328	19620	28469	36021	39486	40675	35345	20447	742	172	149	122	99	3	0	0	0
2.2	2	4	7	14	26	49	90	163	293	550	1027	1910	3533	6259	10195	14217	18638	26597	33210	35800	36376	33168	25975	11900	178	144	117	118	59	10	0	0
3.1	2	4	7	14	26	49	90	164	299	556	1031	1930	3563	6409	10605	14575	19636	28265	36006	39168	39704	34059	19719	4170	189	154	122	87	26	0	0	0
3.2	2	4	7	14	26	48	89	156	291	542	993	1822	3268	5683	8727	11689	17213	23782	29335	33893	36710	37893	35513	11717	415	276	212	185	94	25	5	0
4.1	2	4	7	14	26	49	90	164	300	556	1031	1932	3566	6417	10850	14413	19746	28604	36221	39635	40488	34803	20255	897	178	156	125	96	4	0	0	0
4.2	2	4	7	14	26	48	89	156	291	540	992	1816	3263	5627	8599	11662	17141	23569	29009	33580	36310	36815	33197	16584	470	293	205	178	104	32	5	1
5	2	4	7	14	26	48	90	161	296	546	1012	1865	3405	6025	9486	12695	18219	25638	31659	35695	37940	36914	29310	8581	330	234	191	153	72	11	0	0
6.1	2	4	7	14	26	49	90	164	295	551	1027	1919	3543	6325	10419	14300	19129	27351	34480	37341	37653	32784	23480	9068	185	144	119	105	50	5	0	0
6.2	2	4	7	14	26	48	90	162	297	545	1006	1853	3373	5937	9108	12310	17946	25044	31002	35311	37675	36620	29578	11615	341	247	199	161	86	20	2	0



Slika 4.3 Grafik raspodele čvorova po nivoima za tabelu RIB.3

U tabeli 4.9 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.4 sa rutera RRC12 Frankfurt. Ova RIB tabela sadrži 266659 prefiksa. Analizom tabele, možemo utvrditi da stabla formirana pomoću tehnika 1.1, 1.2, 2.1, 3.1 i 4.1 imaju najmanje dubine (dubina 29). Kod tehnika 2.1 i 4.1 levo podstablo (u odnosu na koren stabla) ima dubinu 28. Stabla najveće dubine formirana su pomoću tehnika 3.2, 4.2, i 6.2 (dubina 32). Najveći stepen simetrije imaju stabla formirana pomoću tehnika 2.1 i 4.1. Najmanji stepen simetrije imaju stabla formirana pomoću tehnika 4.2, 3.2 i 6.2. Transformacija stabla obavlja se sa najmanjim brojem optimizacija pomoću tehnika 6.1 i 2.2, dok je najveći broj optimizacija potreban tehnikama 3.2 i 6.2. Što se tiče vremena potrebnog za formiranje stabla sa prioritetima, možemo uočiti da sve tehnike osim tehnika 6.1 i 6.2 vrše formiranje stabla za približno isto vreme. Tehnike 6.1 i 6.2 u ovom pogledu, ne predstavljaju dobro rešenje.

U tabeli 4.10 prikazan je broj čvorova po nivoima za tabelu RIB.4 sa rutera RRC12 Frankfurt. Analizom tabele 4.10 možemo zaključiti da stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Broj čvorova binarnog stabla najveći je na nivou 24 i iznosi oko 141 hiljadu. Broj čvorova na ovom nivou smanjen je na 764 čvora tehnikom 2.1 odnosno na 914 čvorova tehnikom 4.1. U pogledu broja čvorova po nivoima, tehnike 1.1, 1.2, 3.1 i 6.1 imaju nešto manje povoljnu raspodelu čvorova, što je najizraženije u višim nivoima (veći broj čvorova). Stabla formirana pomoću tehnika 5 i 6.2 nemaju povoljnu raspodelu čvorova po nivoima, dok tehnike 3.2 i 4.2 formiraju stabla sa najnepovoljnijom raspodelom čvorova.

Analizom rezultata kreiranih za RIB.4 tabelu, možemo doći do zaključka da tehnike formiranja stabla sa prioritetima 2.1 i 4.1 predstavljaju najbolje rešenje. Nešto lošije rezultate imaju tehnike 3.1 i 1.2. Tehnike 2.2, 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer vrše formiranje stabla sa nepovoljnom raspodelom čvorova i niskim nivoom simetrije stabla. Takođe, tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje pre svega zbog trajanja formiranja stabla.

Analizom rezultata kreiranih na osnovu tabela RIB.1, RIB.2, RIB.3 i RIB.4 možemo doći do zaključka da za RIB tabele koje sadrže oko 250 hiljada prefiksa, lukap funkcija imaće bolje performanse ako kao strukturu podataka ima stablo sa prioritetima kreirano pomoću bilo koje od navedenih metoda, nego ako ta struktura predstavlja binarno stablo. Međutim, stabla kreirana pomoću tehnika 2.1 i 4.1 predstavljaju najbolje rešenje u pogledu performansi lukap funkcije. Stabla kreirana pomoću ovih tehnika imaju najbolju raspodelu čvorova po nivoima. U poređenju sa drugim stablima sa prioritetima, veći broj čvorova smešten je u nižim nivoima, a manji broj čvorova u višim nivoima. Takva raspodela omogućava da se veći broj pretraga okonča u nižim nivoima stabla. Takođe stabla formirana pomoću ovih tehnika imaju najmanju dubinu i najveći stepen simetrije. Stabla formirana pomoću tehnika 1.1, 1.2 i 3.1 imaju nešto lošiju raspodelu čvorova po nivoima, i manji nivo simetrije stabla. Poređenjem tehnika 1.1 i 1.2 možemo zaključiti da je slučajan izbor kao treći kriterijum bolje rešenje od determinističkog kriterijuma. Stabla formirana pomoću tehnika 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer imaju nepovoljnu raspodelu čvorova po nivoima, i veoma nizak nivo simetrije stabla. Tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje, pre svega jer je vreme trajanja formiranja stabla daleko duže nego što je to slučaj kod ostalih tehnika formiranja stabla.

Tabela 4.9 Rezultati formiranja stabla sa prioritetima za RIB.4 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	685522	260947	424574	418862	266659	107384	159275	418151	242917	199687	242918	69182	104554	72037	/	/
1.1	29	29	29	266659	107384	159274	0	266659	107384	159274	136413	99841	66976	99842	12463	74916	48316	208481	77.069
1.2	29	29	29	266659	107384	159274	0	266659	107384	159274	141811	102299	62060	102300	15142	72016	45297	210143	76.997
2.1	29	28	29	266659	107384	159274	0	266659	107384	159274	160204	107257	52144	107258	21100	65058	41114	216606	77.208
2.2	30	30	30	266659	107383	159275	0	266659	107383	159275	126306	95760	75138	95761	8196	79369	47733	204342	75.827
3.1	29	29	29	266659	107384	159274	0	266659	107384	159274	141729	102315	62028	102316	15107	72102	45229	209959	75.893
3.2	31	31	31	266659	107383	159275	0	266659	107383	159275	111636	84389	97880	84390	7394	69602	46462	228919	75.93
4.1	29	28	29	266659	107384	159274	0	266659	107384	159274	158774	107078	52502	107079	20803	65473	41206	216225	77.778
4.2	31	31	31	266659	107383	159275	0	266659	107383	159275	102176	79531	107596	79532	1798	75936	49862	221247	76.084
5	30	30	30	266659	107383	159275	0	266659	107383	159275	122226	91294	84070	91295	10574	70147	46036	222418	76.436
6.1	30	30	30	266659	107384	159274	0	266659	107384	159274	132475	98166	70326	98167	11157	75853	46399	205969	3087.74
6.2	31	31	30	266659	107383	159275	0	266659	107383	159275	111809	84612	97434	84613	5059	74495	51154	228577	3685.56

Tabela 4.10 Broj čvorova po nivoima za RIB.4 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	26	49	92	169	311	580	1071	2018	3733	6771	12169	20361	23866	36723	53881	67752	82636	101621	119397	141421	2850	2442	1940	1782	1159	473	95	106
p.č.bin.	0	0	0	0	0	0	0	18	9	17	47	148	299	525	1058	10077	4361	7520	15823	18384	18034	22857	23854	139320	835	817	478	900	783	389	0	106
1.1	2	4	7	14	26	49	91	167	306	566	1044	1954	3602	6464	10646	14754	19753	28607	36423	40025	41095	35312	19645	5545	168	149	124	86	30	0	0	0
1.2	2	4	7	14	26	49	91	167	306	566	1044	1954	3601	6469	10716	14688	19766	28654	36464	39993	41213	35325	20624	4354	176	148	119	95	23	0	0	0
2.1	2	4	7	14	26	49	91	167	306	566	1044	1956	3602	6478	10945	14515	19856	28970	36628	40432	42083	36436	21176	764	165	146	119	107	4	0	0	0
2.2	2	4	7	14	26	49	91	165	303	561	1037	1936	3575	6335	10316	14412	18858	26993	33858	36685	37553	34268	26728	12255	173	140	115	124	65	10	0	0
3.1	2	4	7	14	26	49	91	168	306	567	1043	1956	3602	6478	10721	14768	19858	28799	36547	40176	40988	35148	20384	4377	180	159	121	92	27	0	0	0
3.2	2	4	7	14	26	48	91	158	297	547	1006	1841	3303	5750	8841	11838	17457	24181	29869	34630	37833	38985	36691	12102	393	254	192	184	92	20	2	0
4.1	2	4	7	14	26	49	91	168	306	567	1044	1958	3605	6489	10978	14607	19981	29084	36824	40614	41836	35957	20975	914	171	154	124	103	6	0	0	0
4.2	2	4	7	14	26	48	91	158	297	546	1006	1838	3300	5688	8704	11822	17360	23940	29548	34338	37409	37904	34227	17162	452	268	193	177	100	27	2	0
5	2	4	7	14	26	49	91	161	299	551	1023	1899	3448	6088	9581	12863	18495	26042	32171	36542	39133	38098	30301	8838	302	214	177	165	68	6	0	0
6.1	2	4	7	14	26	49	91	166	305	562	1038	1946	3589	6396	10546	14489	19378	27805	35082	38242	38876	33831	24284	9325	183	140	117	108	51	6	0	0
6.2	2	4	7	14	26	48	91	166	302	555	1017	1875	3405	6001	9219	12472	18195	25464	31553	36115	38725	37746	30629	12035	314	224	188	164	84	17	1	0

U tabeli 4.11 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.5 sa rutera RRC05 Vienna. Ova RIB tabela sadrži 494420 prefiksa. Stabla kreirana pomoću tehnika 1.1 i 2.1 imaju najmanju dubinu, koja iznosi 27, pri čemu kod oba stabla levo podstablo (u odnosu na koren stabla) ima dubinu 24. Najveću dubinu (32) ima stablo kreirano pomoću tehnike 4.2. Najveći stepen simetrije imaju stabla kreirana pomoću tehnika 2.1 i 4.1. Stabla kreirana pomoću tehnika 1.2 i 3.1 imaju nešto manji stepen simetrije. Stabla kreirana pomoću tehnika 2.2, 3.2, 5 i 6.2 nemaju povoljan nivo simetrije, dok najniži nivo simetrije stabla ima stablo kreirano pomoću tehnike 4.2. Tehnici 2.2 je potreban najmanji broj optimizacija za transformaciju stabla, dok je tehnici 3.2 potreban najveći broj optimizacija. Iako je broj optimizacija tehnike 3.2 najveći, vreme trajanja formiranja stabla je najmanje. Ovo je posledica činjenice da se proces izbora list čvora razlikuje od tehnike do tehnike, i ima različit broj upita i pristupa memoriji. Vreme trajanja formiranja stabla je veoma slično kod svih tehnika osim kod tehnika 6.1 i 6.2. Ove tehnike ne predstavljaju dobro rešenje u pogledu vremena trajanja formiranja stabla.

U tabeli 4.12 prikazana je raspodela čvorova po nivoima stabala sa prioritetima za tabelu RIB.5 sa rutera RRC05 Vienna. Analizom tabele 4.12 možemo doći do zaključka da stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Ova stabla imaju veći broj čvorova u nižim nivoima a manji broj čvorova u višim nivoima u poređenju sa ostalim tehnikama. Binarno stablo ima najveći broj čvorova na nivou 24 i on iznosi oko 263 hiljada. Na ovom nivou stablo sa prioritetima formirano pomoću tehnike 2.1 ima 2654 čvorova, dok stablo formirano pomoću tehnike 4.1 ima 3138 čvorova. Tehnike formiranja stabla 1.2 i 3.1 predstavljaju dobro rešenje u pogledu raspodele čvorova po nivoima. Stabla formirana pomoću tehnika 3.2 i 4.2 imaju najnepovoljniju raspodelu čvorova po nivoima. Raspodela čvorova po nivoima ilustrovana je na slici 4.4.

Tabela 4.13 prikazuje odnos kumulativnog broja popunjenih čvorova po nivoima i ukupnog broja čvorova, odnosno procenat popunjenih čvorova smeštenih do određenog nivoa. Grafički prikaz ove tabele dat je na slici 4.5. Na osnovu tabele i grafika, možemo primetiti da stabla sa prioritetima formirana pomoću svih tehnika imaju bolju raspodelu čvorova po nivoima od binarnog stabla. Najznačajnija razlika uočljiva je do nivoa 24. Takođe možemo uočiti da je kod binarnog stabla od korena stabla do nivoa 23 smešteno 46.65% popunjenih čvorova, dok je kod nekih stabala sa prioritetima oko 55% popunjenih čvorova smešteno već do nivoa 20.

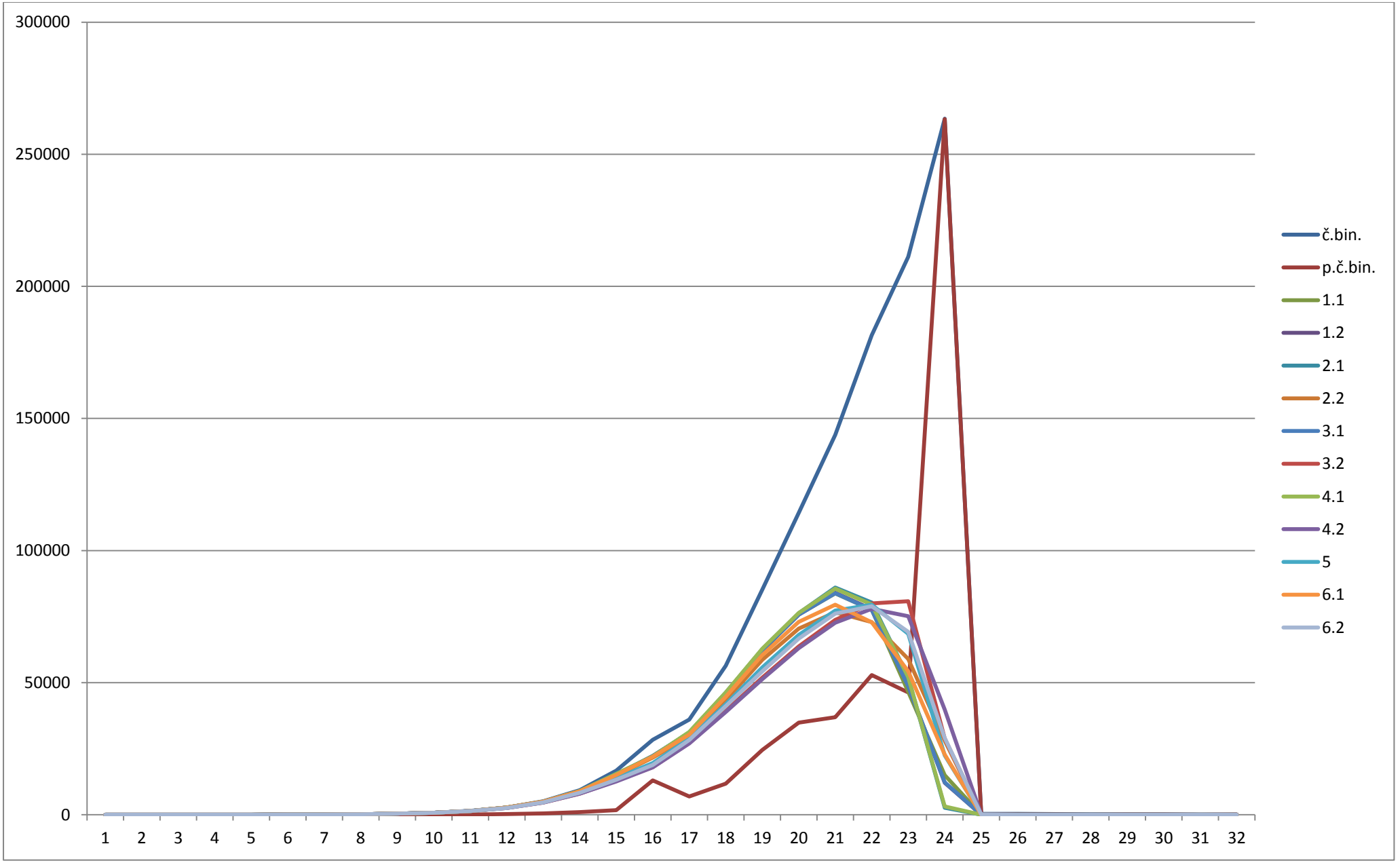
Analizom rezultata kreiranih na osnovu tabele RIB.5 možemo doći do zaključka da za tabelu RIB.5, stablo sa prioritetima kreirano pomoću tehnika 2.1 i 4.1 predstavlja najbolje rešenje. Stablo kreirano pomoću tehnike 2.1 manju dubinu i nešto bolju simetriju nego stablo kreirano pomoću tehnike 4.1, dok stablo kreirano pomoću tehnike 4.1 ima nešto veći broj čvorova u nivoima ispod nivoa 21. U odnosu na ova stabla, stabla formirana pomoću tehnika 1.2 i 3.1 imaju slabije rezultate u pogledu dubine, simetrije stabla i raspodele čvorova, ali ne vrše smanjivanje broja čvorova na nivou 24 u meri u kojoj to postižu tehnike 2.1 i 4.1. Tehnike 6.1 i 6.2 nisu dobro rešenje pre svega zbog vremena potrebnog za formiranje stabla sa prioritetima. Stabla formirana pomoću tehnika 2.2, 3.2 i 5 imaju nepovoljnu raspodelu čvorova po nivoima i veoma nizak nivo simetrije. Stablo formirano pomoću tehnike 4.2 ima najniži nivo simetrije i najnepovoljniju raspodelu čvorova po nivoima.

Tabela 4.11 Rezultati formiranja stabla sa prioritetima za RIB.5 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	1158702	491316	667385	664282	494420	216875	277545	783812	445148	268405	445149	142929	159291	102883	/	/
1.1	27	24	27	494420	216875	277544	0	494420	216875	277544	256439	185497	123425	185498	22840	139818	90632	370164	91.422
1.2	28	24	28	494420	216875	277544	0	494420	216875	277544	268754	191130	112159	191131	29059	133013	83569	374227	90.958
2.1	27	24	27	494420	216875	277544	0	494420	216875	277544	305636	201172	92075	201173	41235	118703	74976	388906	90.313
2.2	30	24	30	494420	216874	277545	0	494420	216874	277545	243421	180475	133469	180476	15837	148802	89139	363309	89.748
3.1	30	26	30	494420	216875	277544	0	494420	216875	277544	268371	190919	112581	190920	28880	133160	83822	373870	90.152
3.2	31	28	31	494420	216874	277545	0	494420	216874	277545	220981	161713	170993	161714	15415	130884	86272	409386	89.651
4.1	29	27	29	494420	216875	277544	0	494420	216875	277544	303055	200819	92781	200820	40725	119370	75124	388177	90.694
4.2	32	28	32	494420	216874	277545	0	494420	216874	277545	205067	152537	189345	152538	4955	142628	92831	393593	89.93
5	31	27	31	494420	216875	277544	0	494420	216875	277544	236210	172385	149649	172386	20229	131928	85681	395460	90.539
6.1	29	24	29	494420	216875	277544	0	494420	216875	277544	252061	183673	127073	183674	20887	141900	86401	365519	9143.2
6.2	31	28	31	494420	216874	277545	0	494420	216874	277545	216109	159097	176225	159098	9195	140708	96123	408578	10802.5

Tabela 4.12 Broj čvorova po nivoima za RIB.5 tabelu

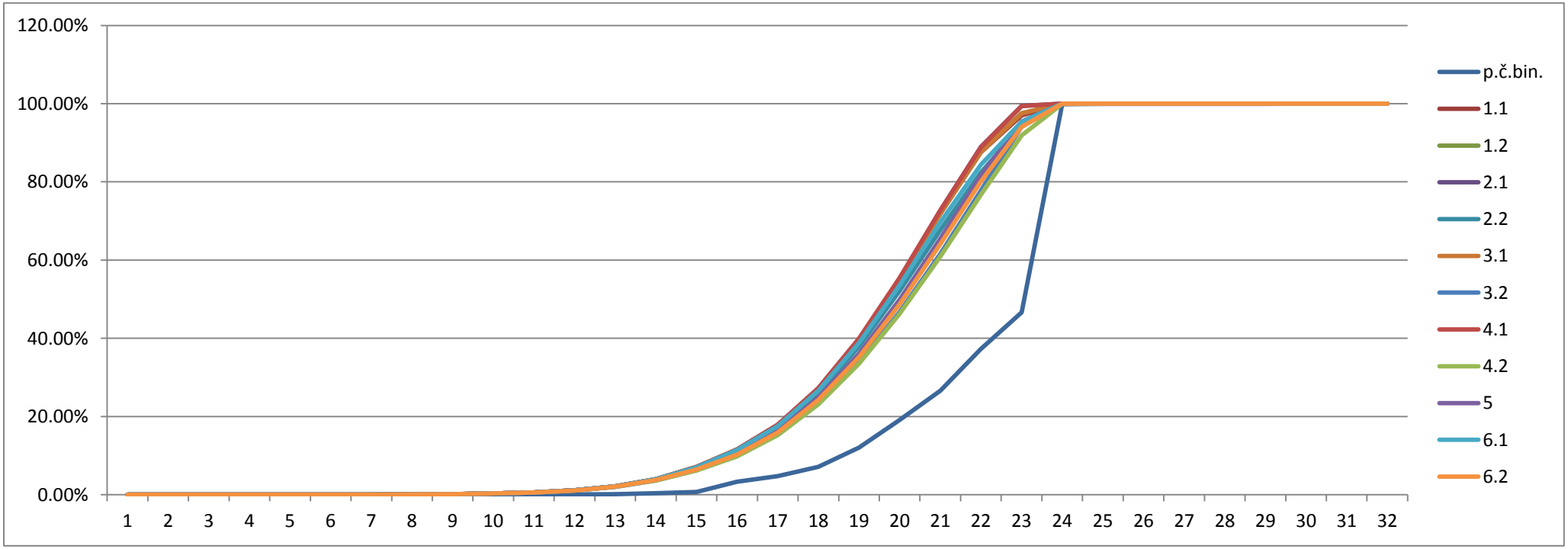
nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	28	56	111	209	398	763	1454	2755	5092	9300	16674	28366	36033	56411	85153	114303	143761	181519	211173	263486	365	311	246	198	180	142	91	96
p.č.bin.	0	0	0	0	0	0	0	16	12	30	91	259	486	973	1713	12957	6950	11738	24466	34895	36884	52900	46254	263276	98	90	73	45	56	60	2	96
1.1	2	4	7	14	28	56	111	208	395	755	1432	2700	4951	8961	15150	22245	30975	45680	62057	75816	83766	77624	46528	14933	9	9	3	0	0	0	0	0
1.2	2	4	7	14	28	56	111	208	395	755	1432	2701	4951	8962	15254	22158	31021	45785	62202	75712	84016	77739	48760	12125	9	8	3	1	0	0	0	0
2.1	2	4	7	14	28	56	111	208	395	755	1432	2701	4952	8978	15471	21991	31189	46212	62647	76290	86025	80304	51972	2654	9	10	2	0	0	0	0	0
2.2	2	4	7	14	28	56	111	207	394	752	1427	2682	4912	8816	14784	21650	29588	43367	58583	70488	76632	72907	58921	28063	6	5	6	3	3	1	0	0
3.1	2	4	7	14	28	56	111	208	395	754	1433	2700	4958	8971	15247	22228	31066	45831	62253	75789	83814	77546	48675	12287	14	12	8	4	3	1	0	0
3.2	2	4	7	14	28	56	110	203	387	729	1370	2522	4587	8005	12836	17998	27162	39322	51869	63682	73723	79955	80796	28880	82	38	19	14	9	6	4	0
4.1	2	4	7	14	28	56	111	208	395	755	1433	2704	4959	8993	15485	22075	31305	46254	62779	76410	85611	79606	52044	3138	14	12	10	5	2	0	0	0
4.2	2	4	7	14	28	56	110	203	387	729	1368	2517	4580	7958	12618	17946	26978	38918	51289	63047	72633	77915	75147	39768	104	41	18	15	9	6	3	1
5	2	4	7	14	28	56	110	206	390	743	1401	2601	4781	8481	13802	19545	28836	41858	55732	68070	77244	79505	68447	22438	48	29	18	12	5	5	1	0
6.1	2	4	7	14	28	56	111	207	394	753	1431	2695	4938	8895	15024	21978	30336	44643	60443	73038	79531	72851	54242	22774	7	5	5	5	2	0	0	0
6.2	2	4	7	14	28	56	111	207	392	740	1391	2565	4701	8322	13249	18818	28257	41126	54224	66696	76134	78930	69253	29058	58	30	22	14	6	3	1	0



Slika 4.4 Grafik raspodele čvorova po nivoima za tabelu RIB.5.

Tabela 4.13 Odnos kumulativnog broja popunjenih čvorova po nivoima i ukupnog broja čvorova za tabelu RIB.5 [%]

nivo:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
p.č.bin.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.4	0.7	3.3	4.8	7.1	12.1	19.1	26.6	37.3	46.6	99.9	99.9	99.9	99.9	100.0	100.0	100.0	100.0	100.0
1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.2	2.2	4.0	7.0	11.5	17.8	27.0	39.6	54.9	71.9	87.6	97.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.2	2.2	4.0	7.1	11.5	17.8	27.1	39.7	55.0	72.0	87.7	97.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.2	2.2	4.0	7.1	11.5	17.9	27.2	39.9	55.3	72.7	88.9	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.1	2.1	3.9	6.9	11.3	17.3	26.1	37.9	52.2	67.7	82.4	94.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.2	2.2	4.0	7.1	11.6	17.8	27.1	39.7	55.0	72.0	87.7	97.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.1	2.0	3.6	6.2	9.9	15.4	23.3	33.8	46.7	61.6	77.8	94.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.2	2.2	4.0	7.1	11.6	17.9	27.3	40.0	55.4	72.7	88.8	99.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.1	2.0	3.6	6.2	9.8	15.3	23.1	33.5	46.3	61.0	76.7	91.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.1	2.1	3.8	6.6	10.6	16.4	24.9	36.1	49.9	65.5	81.6	95.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
6.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.2	2.2	4.0	7.0	11.4	17.6	26.6	38.8	53.6	69.7	84.4	95.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
6.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.6	1.1	2.1	3.7	6.4	10.2	16.0	24.3	35.2	48.7	64.1	80.1	94.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0



Slika 4.5 Grafički prikaz odnosa kumulativnog broja popunjenih čvorova po nivoima i ukupnog broja čvorova za tabelu RIB.5 [%]

U tabeli 4.14 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.6 sa rutera RRC05 Vienna. Ova RIB tabela sadrži 502605 prefiksa. Stabla kreirana pomoću tehnika 1.1, 1.2, 2.1 i 6.1 imaju najmanju dubinu, koja iznosi 26, pri čemu kod sva četiri stabla levo podstablo (u odnosu na koren stabla) ima dubinu 24. Najveću dubinu (32) ima stablo kreirano pomoću tehnike 4.2. Najveći stepen simetrije imaju stabla kreirana pomoću tehnika 2.1 i 4.1. Stabla kreirana pomoću tehnika 1.2 i 3.1 imaju nešto manji stepen simetrije. Stabla kreirana pomoću tehnika 2.2, 3.2, 5 i 6.2 nemaju povoljan nivo simetrije, dok najniži nivo simetrije stabla ima stablo kreirano pomoću tehnike 4.2. Tehnici 2.2 je potreban najmanji broj optimizacija za transformaciju stabla, dok je tehnici 3.2 potreban najveći broj optimizacija. Vreme trajanja formiranja stabla je veoma slično kod svih tehnika osim kod tehnika 6.1 i 6.2. Ove tehnike ne predstavljaju dobro rešenje zbog veoma velikog vremena potrebnog za forimranje stabla sa prioritetima.

U tabeli 4.15 prikazana je raspodela čvorova po nivoima stabala sa prioritetima za tabelu RIB.6 sa rutera RRC05 Vienna. Analizom tabele 4.15 možemo doći do zaključka da iako dubina stabla kod stabala sa prioritetima nije značajno smanjena, raspodela čvorova je znatno bolja nego što je to slučaj kod binarnog stabla. Stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Binarno stablo ima najveći broj čvorova na nivou 24 i on iznosi oko 267 hiljada. Na ovom nivou stablo sa prioritetima formirano pomoću tehnike 2.1 ima 2907 čvorova, dok stablo formirano pomoću tehnike 4.1 ima 3432 čvorova. Tehnike formiranja stabla 1.2 i 3.1 predstavljaju dobro rešenje u pogledu raspodele čvorova po nivoima, ali ne vrše smanjivanje broja čvorova na nivou 24 u meri u kojoj to postižu tehnike 2.1 i 4.1. Stabla formirana pomoću tehnika 3.2 i 4.2 imaju najnepovoljniju raspodelu čvorova po nivoima.

Analizom rezultata kreiranih za RIB.6 tabelu, možemo doći do zaključka da tehnike formiranja stabla sa prioritetima 2.1 i 4.1 predstavljaju najbolje rešenje. Nešto lošije rezultate imaju tehnike 3.1 i 1.2. Tehnike 2.2, 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer vrše formiranje stabla sa nepovoljnom raspodelom čvorova i niskim nivoom simetrije stabla. Takođe, tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje pre svega zbog vremena potrebnog za formiranje stabla sa prioritetima.

Tabela 4.14 Rezultati formiranja stabla sa prioritetima za RIB.6 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	1E+06	498340	678226	673961	502605	219717	282888	796050	452251	272064	452252	145301	161650	104333	/	/
1.1	26	24	26	502605	219717	282887	0	502605	219717	282887	260935	188607	125390	188608	23231	142146	92132	375824	94.215
1.2	26	24	26	502605	219717	282887	0	502605	219717	282887	273773	194383	113838	194384	29627	135130	84862	379988	93.429
2.1	26	24	26	502605	219717	282887	0	502605	219717	282887	310548	204414	93776	204415	41775	120865	76420	394902	92.701
2.2	27	24	27	502605	219716	282888	0	502605	219716	282888	247499	183521	135562	183522	16067	151388	90631	368975	92.533
3.1	29	24	29	502605	219717	282887	0	502605	219717	282887	273416	194333	113938	194334	29612	135110	84880	379688	92.914
3.2	31	27	31	502605	219716	282888	0	502605	219716	282888	225430	164674	173256	164675	15993	132689	87545	416037	92.703
4.1	29	24	29	502605	219717	282887	0	502605	219717	282887	307982	204075	94454	204076	41276	121524	76507	394120	92.645
4.2	32	27	32	502605	219716	282888	0	502605	219716	282888	208731	155168	192268	155169	5091	144987	94367	399639	92.628
5	30	26	30	502605	219717	282887	0	502605	219717	282887	239772	175113	152378	175114	20512	134090	87297	401511	92.698
6.1	26	24	26	502605	219717	282887	0	502605	219717	282887	256589	186848	128908	186849	21333	144183	87802	371148	9515.36
6.2	31	27	31	502605	219716	282888	0	502605	219716	282888	219833	161792	179020	161793	9413	142967	97630	414863	11283

Tabela 4.15 Broj čvorova po nivoima za RIB.6 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	28	56	111	210	403	769	1463	2773	5143	9409	16876	28723	36592	57153	86623	116382	146192	184261	214236	267377	436	368	303	224	179	97	73	79
p.č.bin.	0	0	0	0	0	0	0	16	13	30	91	259	496	990	1721	13021	7032	11712	24848	35607	37745	54066	47258	267131	112	90	97	68	92	31	0	79
1.1	2	4	7	14	28	56	111	210	400	760	1441	2727	5004	9068	15338	22496	31316	46359	63135	77130	85196	78948	47445	15402	4	3	0	0	0	0	0	0
1.2	2	4	7	14	28	56	111	210	400	760	1441	2727	5006	9070	15424	22431	31338	46483	63274	77123	85313	79141	49692	12542	5	2	0	0	0	0	0	0
2.1	2	4	7	14	28	56	111	210	400	760	1441	2728	5006	9085	15664	22258	31487	46927	63732	77611	87380	81767	53012	2907	6	1	0	0	0	0	0	0
2.2	2	4	7	14	28	56	111	209	398	756	1438	2710	4971	8928	14973	21867	29908	44055	59664	71666	77897	74216	59913	28802	5	4	2	0	0	0	0	0
3.1	2	4	7	14	28	56	111	210	400	760	1441	2726	5014	9082	15443	22482	31385	46524	63407	77071	85139	78983	49597	12697	10	5	2	2	2	0	0	0
3.2	2	4	7	14	28	56	111	205	392	738	1385	2546	4640	8128	12966	18205	27506	39887	52838	64785	74978	81123	82588	29316	90	35	12	8	4	3	4	0
4.1	2	4	7	14	28	56	111	210	400	760	1442	2728	5015	9104	15683	22338	31595	46999	63836	77770	86962	81089	52998	3432	11	4	2	2	2	0	0	0
4.2	2	4	7	14	28	56	111	205	392	738	1385	2543	4633	8069	12776	18142	27308	39458	52234	64158	73795	79087	76499	40764	127	37	12	8	5	3	3	1
5	2	4	7	14	28	56	111	208	395	748	1417	2633	4819	8581	13988	19787	29081	42538	56740	69324	78469	80896	69554	23107	51	26	8	4	4	4	0	0
6.1	2	4	7	14	28	56	111	210	398	758	1440	2721	4990	8996	15230	22210	30682	45286	61526	74354	80825	74126	55269	23353	6	2	0	0	0	0	0	0
6.2	2	4	7	14	28	56	111	209	396	745	1400	2586	4751	8426	13425	19045	28606	41749	55173	67865	77293	80224	70634	29738	65	27	12	6	4	2	1	0

U tabeli 4.16 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.7 sa rutera RRC10 Milan. Ova RIB tabela sadrži 495385 prefiksa. Stabla kreirana pomoću tehnika 1.1, 1.2, 2.1, 3.1 i 6.1 imaju najmanju dubinu, koja iznosi 28. Levo podstablo (u odnosu na koren stabla) stabla kreiranog pomoću tehnika 1.1 i 1.2 ima dubinu 26, dok kod stabla kreiranog pomoću tehnike 2.1 ona iznosi 27. Kod stabla kreiranog pomoću tehnike 3.1 desno podstablo ima dubinu 27, dok kod stabla kreiranog pomoću tehnike 4.1 ona iznosi 26. Najveću dubinu (32) imaju stabla kreirana pomoću tehnika 3.2 i 4.2. Najveći stepen simetrije imaju stabla kreirana pomoću tehnika 2.1 i 4.1. Stabla kreirana pomoću tehnika 1.2 i 3.1 imaju nešto manju simetriju. Stabla kreirana pomoću tehnika 2.2, 3.2, 5 i 6.2 nemaju povoljan nivo simetrije, dok najniži nivo simetrije stabla ima stablo kreirano pomoću tehnike 4.2. Tehnici 2.2 je potreban najmanji broj optimizacija za transformaciju stabla, dok je tehnici 3.2 potreban najveći broj optimizacija. Vreme trajanja formiranja stabla je veoma slično kod svih tehnika osim kod tehnika 6.1 i 6.2. Ove tehnike ne predstavljaju dobro rešenje zbog veoma velikog vremena potrebnog za formiranje stabla sa prioritetima.

U tabeli 4.17 prikazana je raspodela čvorova po nivoima stabala sa prioritetima za tabelu RIB.7 sa rutera RRC10 Milan. Analizom tabele 4.17 možemo doći do zaključka da stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Binarno stablo ima najveći broj čvorova na nivou 24 i on iznosi oko 263 hiljade. Na ovom nivou stablo sa prioritetima formirano pomoću tehnike 2.1 ima 2759 čvorova, dok stablo formirano pomoću tehnike 4.1 ima 3264 čvorova. Tehnike formiranja stabla 1.2 i 3.1 predstavljaju dobro rešenje u pogledu raspodele čvorova po nivoima, ali ne vrše smanjivanje broja čvorova na nivou 24 u meri u kojoj to postižu tehnike 2.1 i 4.1. Stabla formirana pomoću tehnika 3.2 i 4.2 imaju najnepovoljniju raspodelu čvorova po nivoima. Grafički prikaz raspodele čvorova dat je na slici 4.6.

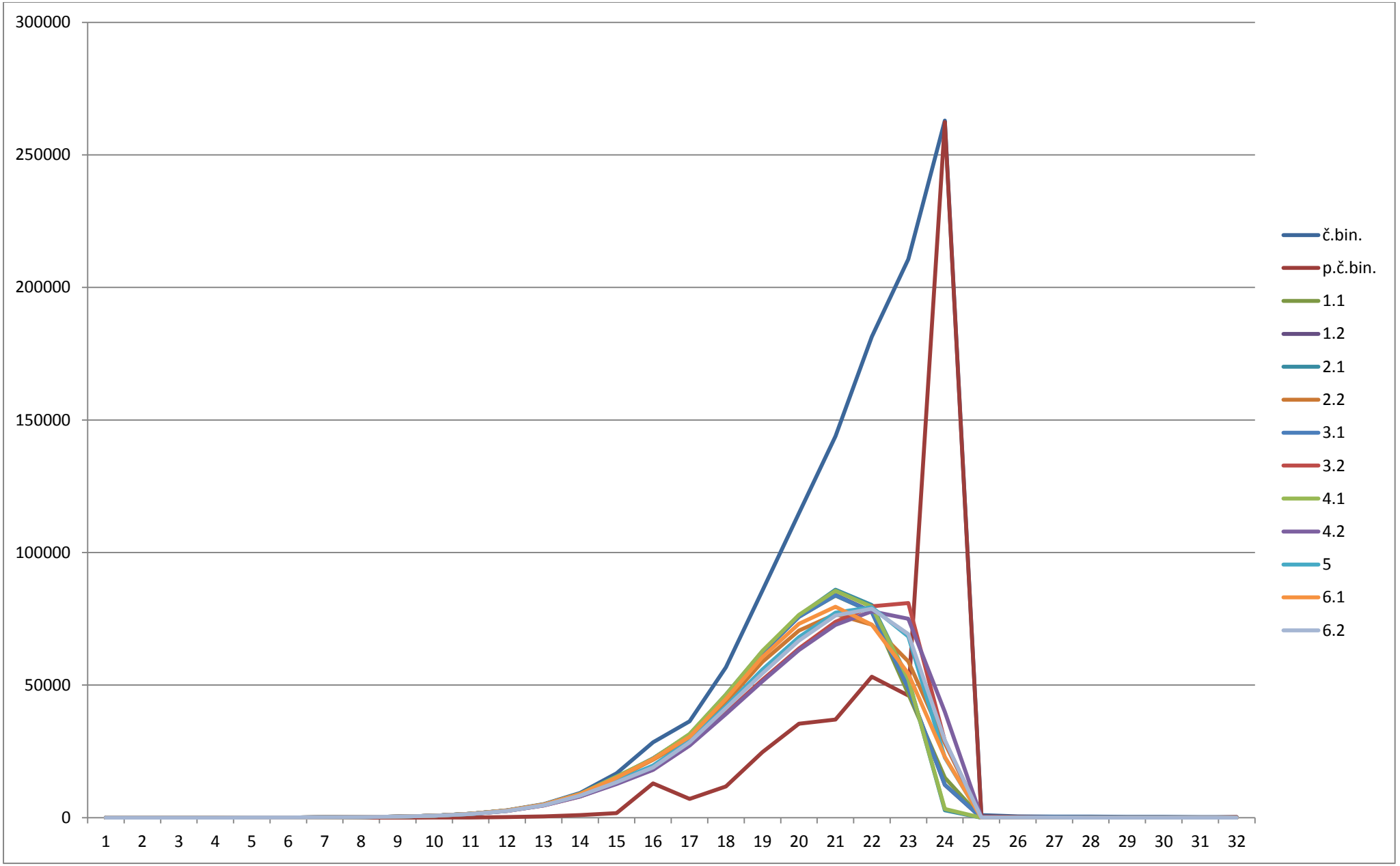
Analizom rezultata kreiranih za RIB.7 tabelu, možemo doći do zaključka da tehnike formiranja stabla sa prioritetima 2.1 i 4.1 predstavljaju najbolje rešenje. Nešto lošije rezultate imaju tehnike 3.1 i 1.2. Tehnike 2.2, 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer vrše formiranje stabla sa nepovoljnom raspodelom čvorova i niskim nivoom simetrije stabla. Takođe, tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje pre svega zbog vremena potrebnog za formiranje stabla sa prioritetima.

Tabela 4.16 Rezultati formiranja stabla sa prioritetima za RIB.7 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	1E+06	494498	666470	665583	495385	218179	277206	783130	445741	269486	445742	143049	159644	103102	/	/
1.1	28	26	28	495385	218179	277205	0	495385	218179	277205	257187	185897	123590	185898	22905	140088	90883	370576	83.804
1.2	28	26	28	495385	218179	277205	0	495385	218179	277205	269330	191530	112324	191531	28995	133541	83855	374583	83.78
2.1	28	27	28	495385	218179	277205	0	495385	218179	277205	306450	201602	92180	201603	41370	118863	75149	389387	83.022
2.2	30	26	30	495385	218178	277206	0	495385	218178	277206	244197	180944	133496	180945	16021	148903	89240	363738	82.884
3.1	28	28	27	495385	218179	277205	0	495385	218179	277205	269196	191505	112374	191506	29081	133344	83817	374333	82.995
3.2	32	31	32	495385	218178	277206	0	495385	218178	277206	221431	162144	171096	162145	15473	131199	86498	409913	82.65
4.1	28	28	26	495385	218179	277205	0	495385	218179	277205	303767	201239	92906	201240	40814	119612	75317	388616	82.702
4.2	32	31	32	495385	218178	277206	0	495385	218178	277206	205138	152900	189584	152901	5029	142843	93002	393991	82.774
5	30	30	30	495385	218178	277206	0	495385	218178	277206	236574	172743	149898	172744	20344	132056	85799	395618	83.83
6.1	29	28	29	495385	218179	277205	0	495385	218179	277205	252688	184096	127192	184097	21002	142093	86564	365897	9259.2
6.2	31	31	31	495385	218178	277206	0	495385	218178	277206	216631	159529	176326	159530	9305	140920	96268	408991	11386.1

Tabela 4.17 Broj čvorova po nivoima za RIB.7 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	28	56	111	209	398	763	1454	2753	5086	9297	16699	28461	36325	56754	85655	114812	143840	181422	210629	262970	934	520	360	360	314	281	218	232
p.č.bin.	0	0	0	0	0	0	0	16	12	30	90	258	486	974	1721	12970	7039	11788	24699	35413	37046	53166	46112	262399	486	193	37	73	62	81	2	232
1.1	2	4	7	14	28	56	111	209	395	755	1432	2700	4949	8967	15192	22368	31166	45828	62152	75837	83743	77648	46684	15112	13	9	2	1	0	0	0	0
1.2	2	4	7	14	28	56	111	209	395	755	1432	2701	4948	8968	15278	22300	31224	45906	62321	75730	84008	77748	48846	12368	14	8	2	1	0	0	0	0
2.1	2	4	7	14	28	56	111	209	395	755	1432	2701	4950	8985	15506	22123	31379	46367	62754	76277	85980	80241	52324	2759	12	9	3	1	0	0	0	0
2.2	2	4	7	14	28	56	110	207	394	753	1428	2685	4914	8825	14822	21810	29806	43556	58747	70607	76717	72817	58935	28110	17	7	2	1	2	1	0	0
3.1	2	4	7	14	28	56	111	209	395	755	1432	2701	4954	8977	15278	22361	31287	45999	62416	75786	83834	77557	48780	12402	19	14	5	1	0	0	0	0
3.2	2	4	7	14	28	56	110	203	387	729	1369	2519	4588	8023	12891	18142	27350	39463	52053	63789	73846	79723	80916	28927	121	49	30	20	13	9	2	1
4.1	2	4	7	14	28	56	111	209	395	755	1433	2704	4957	8999	15521	22212	31526	46436	62880	76468	85585	79590	52193	3264	15	14	5	1	0	0	0	0
4.2	2	4	7	14	28	56	110	203	387	730	1369	2515	4578	7970	12667	18078	27186	39062	51436	63191	72697	77828	75043	39886	210	50	28	23	13	9	3	1
5	2	4	7	14	28	56	111	206	391	745	1406	2602	4765	8481	13848	19730	29023	42018	55925	68099	77316	79416	68354	22646	92	38	28	19	8	6	0	0
6.1	2	4	7	14	28	56	111	206	394	753	1432	2695	4936	8897	15059	22104	30526	44788	60578	73130	79565	72832	54327	22908	17	10	2	2	1	0	0	0
6.2	2	4	7	14	28	56	111	208	392	740	1391	2562	4698	8337	13313	18939	28425	41311	54376	66777	76217	78854	69248	29174	93	41	27	17	15	5	2	0



Slika 4.6 Grafik raspodele čvorova po nivoima za tabelu RIB.7.

U tabeli 4.18 prikazani su rezultati formiranja stabala sa prioritetima za tabelu RIB.8 sa rutera RRC10 Milan. Ova RIB tabela sadrži 503158 prefiksa. Stabla kreirana pomoću tehnika 2.1 i 4.1 imaju najmanju dubinu, koja iznosi 31. Desno podstablo (u odnosu na koren stabla) stabla kreiranog pomoću tehnike 2.1 ima dubinu 28, dok kod stabla kreiranog pomoću tehnike 4.1 ona iznosi 26. Stabla kreirana pomoću svih preostalih tehnika imaju dubinu 32, što ukazuje na nepovoljnu raspodelu popunjenih i praznih čvorova u binarnom stablu, naročito u levom podstablu. Naime, ako jedan list čvor dubine 32 nema praznih predaka, tom stablu nije moguće smanjiti dubinu nijednom od navedenih tehnika. Najveći stepen simetrije imaju stabla kreirana pomoću tehnika 2.1 i 4.1. Stabla kreirana pomoću tehnika 1.2 i 3.1 imaju nešto manju simetriju. Stabla kreirana pomoću tehnika 2.2 i 6.2 nemaju povoljan nivo simetrije, dok najniži nivo simetrije stabla ima stablo kreirano pomoću tehnike 4.2. Tehnici 2.2 je potreban najmanji broj optimizacija za transformaciju stabla, dok je tehnici 3.2 potreban najveći broj optimizacija. Vreme trajanja formiranja stabla sa prioritetima je veoma slično kod svih tehnika osim kod tehnika 6.1 i 6.2 kojima je potrebno znatno više vremena za formiranje stabla.

U tabeli 4.19 prikazana je raspodela čvorova po nivoima stabala sa prioritetima za tabelu RIB.8 sa rutera RRC10 Milan. Analizom tabele 4.19 možemo doći do zaključka da stabla formirana pomoću tehnika 2.1 i 4.1 imaju najbolju raspodelu čvorova po nivoima. Binarno stablo ima najveći broj čvorova na nivou 24 i on iznosi oko 266 hiljada. Na ovom nivou stablo sa prioritetima formirano pomoću tehnike 2.1 ima 2958 čvorova, dok stablo formirano pomoću tehnike 4.1 ima 3484 čvorova. Tehnike formiranja stabla 1.2 i 3.1 predstavljaju dobro rešenje u pogledu raspodele čvorova po nivoima, ali ne vrše smanjivanje broja čvorova na nivou 24 u meri u kojoj to postižu tehnike 2.1 i 4.1. Stabla formirana pomoću tehnika 3.2 i 4.2 imaju najnepovoljniju raspodelu čvorova po nivoima.

Analizom rezultata kreiranih za RIB.8 tabelu, možemo doći do zaključka da tehnike formiranja stabla sa prioritetima 2.1 i 4.1 predstavljaju najbolje rešenje. Nešto lošije rezultate imaju tehnike 3.1 i 1.2. Tehnike 2.2, 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer vrše formiranje stabla sa nepovoljnom raspodelom čvorova i niskim nivoom simetrije stabla. Takođe, tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje pre svega zbog vremena potrebnog za formiranje stabla sa prioritetima.

Tabela 4.18 Rezultati formiranja stabla sa prioritetima za RIB.8 tabelu

	dubina	dubina levo	dubina desno	br. čvorova	br. čv. levo	br. čv. desno	br. pr. čv.	br. pop. čv.	br. pop. čv. levo	br. pop. čv. desno	br. čv. sa sim. dub.	br. čv. sa sim. decom	br. čv. sa asim. decom	br. listova	br. čv. sa sim. list.	br. čv. sa asim. list.	br. čv. sa jed. listom	br. opt.	trajanje form. (s)
bin.	32	32	32	1E+06	500572	677031	674445	503158	220614	282544	795760	452572	272459	452573	145422	161729	104364	/	/
1.1	32	32	28	503158	220614	282543	0	503158	220614	282543	261406	188846	125465	188847	23292	142263	92269	375985	84.877
1.2	32	32	27	503158	220614	282543	0	503158	220614	282543	274021	194624	113909	194625	29631	135363	85036	380081	84.757
2.1	31	31	28	503158	220614	282543	0	503158	220614	282543	311262	204698	93761	204699	41944	120811	76447	395115	84.842
2.2	32	32	30	503158	220613	282544	0	503158	220613	282544	248024	183804	135549	183805	16192	151421	90729	369058	83.92
3.1	32	32	27	503158	220614	282543	0	503158	220614	282543	273676	194485	114187	194486	29509	135468	85177	379820	84.24
3.2	32	32	30	503158	220613	282544	0	503158	220613	282544	225303	164739	173679	164740	15745	133250	87910	415886	83.895
4.1	31	31	26	503158	220614	282543	0	503158	220614	282543	308466	204306	94545	204307	41329	121649	76647	394292	84.02
4.2	32	32	30	503158	220613	282544	0	503158	220613	282544	208957	155460	192237	155461	5147	145167	94503	399839	84.085
5	32	32	29	503158	220614	282543	0	503158	220614	282543	240066	175273	152611	175274	20486	134302	87337	401393	84.245
6.1	32	32	29	503158	220613	282544	0	503158	220613	282544	256984	187113	128931	187114	21379	144356	87923	371262	9971.77
6.2	32	32	30	503158	220613	282544	0	503158	220613	282544	220126	162044	179069	162045	9469	143107	97723	414995	11827.5

Tabela 4.19 Broj čvorova po nivoima za RIB.8 tabelu

nivo:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
č.bin.	2	4	7	14	28	56	111	210	402	766	1460	2769	5136	9403	16892	28802	36857	57472	87043	116767	146048	183877	213666	266772	762	529	365	357	306	264	209	247
p.č.bin.	0	0	0	0	0	0	0	16	13	30	91	259	496	990	1722	13032	7109	11782	25044	36114	37870	54090	47167	266304	305	200	41	79	75	80	2	247
1.1	2	4	7	14	28	56	111	210	399	758	1440	2727	5002	9072	15371	22603	31496	46484	63217	77064	85029	78919	47574	15498	17	16	11	4	6	10	6	2
1.2	2	4	7	14	28	56	111	210	399	758	1440	2727	5003	9074	15456	22534	31526	46607	63383	76995	85243	79090	49697	12721	17	18	10	3	6	10	7	1
2.1	2	4	7	14	28	56	111	210	399	758	1440	2728	5004	9088	15694	22363	31675	47060	63810	77546	87177	81729	53225	2958	17	19	7	4	6	12	6	0
2.2	2	4	7	14	28	56	110	209	398	756	1436	2712	4973	8934	15002	21996	30098	44220	59812	71678	77777	74087	59949	28822	21	13	6	4	7	9	11	6
3.1	2	4	7	14	28	56	111	210	399	758	1440	2728	5009	9086	15472	22585	31580	46687	63434	77088	84959	78948	49725	12734	27	24	12	4	6	11	8	1
3.2	2	4	7	14	28	56	110	204	391	737	1384	2543	4635	8142	13013	18344	27663	40042	52882	64846	74928	80963	82316	29608	131	57	29	19	17	18	18	6
4.1	2	4	7	14	28	56	111	210	399	758	1441	2728	5012	9106	15712	22455	31804	47153	63928	77713	86791	81034	53120	3484	23	25	9	4	6	12	8	0
4.2	2	4	7	14	28	56	110	204	391	737	1383	2542	4631	8081	12818	18270	27478	39603	52330	64213	73757	78964	76403	40780	187	57	26	21	17	17	17	9
5	2	4	7	14	28	56	111	207	395	745	1405	2628	4813	8609	14011	19832	29423	42698	56811	69174	78444	80706	69495	23320	90	38	24	16	15	15	17	4
6.1	2	4	7	14	28	56	111	208	398	757	1439	2721	4988	8999	15256	22322	30862	45424	61616	74342	80721	74061	55345	23400	19	16	10	4	5	8	9	5
6.2	2	4	7	14	28	56	111	209	394	744	1398	2584	4750	8439	13475	19159	28765	41891	55285	67859	77227	80095	70621	29799	94	46	29	18	17	15	15	7

Analizom rezultata kreiranih na osnovu svih 8 tabela možemo doći do zaključka da će lukap funkcija imaće bolje performanse ako kao strukturu podataka ima stablo sa prioritetima kreirano pomoću bilo koje od navedenih metoda, nego ako ta struktura predstavlja binarno stablo. Međutim, stabla kreirana pomoću tehnika 2.1 i 4.1 predstavljaju najbolje rešenje u pogledu performansi lukap funkcije. Stabla kreirana pomoću ovih tehnika imaju najbolju raspodelu čvorova po nivoima. U poređenju sa drugim stablima sa prioritetima, veći broj čvorova smešten je u nižim nivoima, a manji broj čvorova u višim nivoima. Takva raspodela omogućava da se veći broj pretraga okonča u nižim nivoima stabla. Takođe stabla formirana pomoću ovih tehnika imaju najmanju dubinu i najveći stepen simetrije. Tehnike 2.1 i 4.1 imaju veoma slične rezultate, međutim ako ih uporedimo, možemo primetiti da stabla formirana pomoću tehnike 2.1 imaju veći stepen simetrije, i manje čvorova u višim nivoima stabla, dok stabla formirana pomoću tehnike 4.1 imaju više čvorova u nižim nivoima stabla, što takođe povoljno utiče na performanse lukap tabele. Takođe, tehnici 4.1 je potreban manji broj optimizacija za formiranje stabla sa prioritetima. Stabla formirana pomoću tehnika 1.1, 1.2 i 3.1 nešto lošiju raspodelu čvorova po nivoima, i manji nivo simetrije stabla. Poređenjem tehnika 1.1 i 1.2 možemo zaključiti da je slučajan izbor kao treći kriterijum bolje rešenje od determinističkog kriterijuma. Stabla formirana pomoću tehnika 3.2, 4.2 i 5 ne predstavljaju dobro rešenje jer imaju nepovoljnu raspodelu čvorova po nivoima, i veoma nizak nivo simetrije stabla. Tehnike 6.1 i 6.2 ne predstavljaju dobro rešenje, pre svega jer je vreme trajanja formiranja stabla daleko duže nego što je to slučaj kod ostalih tehnika formiranja stabla. Ovo može predstavljati veliki problem kada dođe do promene u topologiji mreže, odnosno do ažuriranja RIB tabele.

5. ZAKLJUČAK

Velikim rastom broja uređaja koji imaju pristup Internetu dolazi do ogromnog rasta Internet saobraćaja. Zbog toga, ruteri se suočavaju sa ogromnim povećanjem lukap tabela i brzinom pristizanja paketa. Od rutera se zahteva izvršavanje IP lukap funkcije najvećom mogućom brzinom. IP lukap funkcija može koristiti binarno stablo kao vrlo jednostavnu strukturu podataka. Međutim, sa porastom lukap tabele binarno stablo zahteva velike memorijske resurse zbog velikog broja praznih čvorova koji ne nose korisnu informaciju. Neminovan je i prelaz na IPv6 adrese, što dodatno povećava prostor za pretragu. Takođe, IP lukap funkcija koja koristi binarno stablo kao strukturu podataka vrši veliki broj pristupa memoriji u procesu donošenja odluke o prosleđivanju. Broj pristupa memoriji veoma je bitan obzirom da predstavlja najsporiji proces u procesu donošenja odluke o prosleđivanju. Stablo sa prioritetima transformiše klasično binarno stablo u binarno stablo koje sadrži samo popunjene čvorove. Transformacija se postiže pomeranjem listova stabla u prazne čvorove, do eliminacije svih praznih čvorova u stablu. Pomeranjem popunjenih čvorova iz viših nivoa u niže nivoe stabla, formira se stablo u kome se više pretraga može okončati u nižim nivoima stabla.

U okviru ovog rada, predstavljeno je jedanaest tehnika formiranja stabla sa prioritetima. Tehnike se razlikuju po kriterijumima izbora list čvora koji se premešta u prazan čvor. Analizom rezultata kreiranih na osnovu 8 RIB tabela došli smo do zaključka da će lukap funkcija koja kao strukturu podataka ima stablo sa prioritetima kreirano pomoću bilo koje od navedenih tehnika, imati značajno bolje performanse nego lukap funkcija koja kao strukturu podataka ima binarno stablo. Tehnike formiranja stabla sa prioritetima označene u radu kao tehnika 2.1 i tehnika 4.1, izdvajaju se po najboljim rezultatima. Tehnika 2.1 predstavlja tehniku formiranja stabla sa prioritetima u kojoj se izbor list čvora vrši prema kriterijumima: veća dubina, veći broj čvorova, nasumičan izbor. Tehnika 4.1 predstavlja tehniku formiranja stabla sa prioritetima u kojoj se izbor list čvora vrši prema kriterijumima: veći broj čvorova, veći broj praznih čvorova, nasumičan izbor. Kriterijumi su navedeni počev od kriterijuma najvišeg prioriteta, a izbor list čvora prema kriteriumima vrši se prilikom kretanja kroz stablo. Stabla kreirana pomoću ovih tehnika imaju najbolju raspodelu čvorova po nivoima. U poređenju sa stablima sa prioritetima formiranih pomoću drugih predloženih tehnika, veći broj čvorova smešten je u nižim nivoima, a manji broj čvorova u višim nivoima. Takva raspodela omogućava da se veći broj pretraga okonča u nižim nivoima stabla. Takođe stabla formirana pomoću tehnika 2.1 i 4.1 imaju najmanju dubinu i najveći stepen simetrije.

U odnosu na binarna stabla, stabla sa prioritetima omogućavaju manji broj pristupa memoriji, zbog manje dubine i bolje raspodele popunjenih čvorova po nivoima. Takođe, kod lukap funkcije koja za strukturu podataka koristi stablo sa prioritetima, pretraga se okončava prilikom pronalaženja prvog rešenja, što takođe smanjuje broj pristupa memoriji. Međutim, problem višestrukog pristupa memoriji je i dalje prisutan, jer u najgorem slučaju tokom pretrage se mora ići do list čvora stabla.

LITERATURA

[1] Zoran Čiča, "Implementacija funkcija paketskog procesiranja u Internet ruterima velikog kapaciteta", doktorska teza

[2] Hyesook Lim, Member, IEEE, Changhoon Yim, Member, IEEE, and Earl E. Swartzlander, Jr., Fellow, IEEE, "Priority Tries for IP Address Lookup", June 2010.

[3] Code::Blocks, http://www.codeblocks.org

[4] A. Smiljanić, Z. Čiča, „A Comparative Review of Scalable Lookup Algorithms for IPv6," Computer Networks, vol.56(13), pp. 3040-3054, September 2012.

[5] RIPE Network Coordination Centre, http://www.ripe.net/data-tools/stats/ris/ris-raw-data