

ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU



KONFIGURISANJE OSPF-A U MREŽI SA JUNIPER RUTERIMA

–Diplomski rad –

Kandidat:

Nevena Krunić 2010/407

Mentor:

doc. dr Zoran Čiča

Beograd, Septembar 2016.

SADRŽAJ

SADRŽAJ.....	2
1. UVOD.....	3
2. PROTOKOLI RUTIRANJA.....	4
2.1. KLASIFIKACIJA INTERNIH PROTOKOLA RUTIRANJA.....	4
3. VISOKA RASPOLOŽIVOST I KONVERGENCIJA MREŽE	5
4. OSPF	6
5. FAST HELLO MEHANIZMI I BFD	7
5.1. KONCEPT <i>FAST-HELLO</i> MEHANIZAMA.....	7
5.2. BFD	7
5.2.1. <i>Konfiguracija na Cisco uređajima</i>	9
5.2.2. <i>Konfiguracija na Juniper mrežnim uređajima</i>	10
6. TOPOLOGIJA TEST MREŽE	12
6.1. SIMULACIJA TESTNE MREŽE.....	13
6.1.1. <i>GNS3 Simulator</i>	13
6.1.2. <i>Alat Virtualbox</i>	14
6.1.3. <i>Kreiranje testne topologije</i>	14
6.2. KONFIGURACIJA MREŽNIH UREĐAJA.....	23
6.2.1. <i>Konfiguracija interfejsa</i>	23
6.2.2. <i>Konfiguracija OSPF protokola</i>	23
6.2.3. <i>Konfiguracija BFD protokola</i>	24
7. TESTIRANJE KONFIGURACIJE.....	26
7.1. SLUČAJ PREKIDA JEDNOG OD LINKOVA ILI PADA POJEDINAČNOG INTERFEJSA	28
7.2. PRESTANAK RADA JEDNOG OD RUTERA R1, R2, R4, R6 I R7	31
7.3. PRESTANAK RADA JEDNOG OD ABR (<i>AREA BORDER ROUTER</i>) RUTERA	33
8. ZAKLJUČAK.....	34
LITERATURA.....	35
A. PRILOG A.....	36
A.1. KONFIGURACIJA RUTERA R1	36
A.2. KONFIGURACIJA RUTERA R2	38
A.3. KONFIGURACIJA RUTERA R3	39
A.4. KONFIGURACIJA RUTERA R4	43
A.5. KONFIGURACIJA RUTERA R5	44
A.6. KONFIGURACIJA RUTERA R6	47
A.7. KONFIGURACIJA RUTERA R7	48
B. ADRESNA ŠEMA TESTNE TOPOLOGIJE.....	52

1. UVOD

Dostupnost, brzina i podrška za vremenski osetljive servise su postali imperativ kako u IT industriji tako i u medicini, ekonomiji i drugim industrijskim sferama. Podrška za servise kao što su *voice* i *real-time* video prenos koji su osetljivi na gubitak paketa, greške u transmisiji, kašnjenje i džiter koristeći IP (*Internet Protocol*) kao tehnologiju, koja je upravo projektovana tako da bude tolerantna na navedene stavke.

Sa tehnološkim razvojem i sve većim akcentom na IT industriju, javlja se sve veća konkurencija između proizvođača kako same mrežne opreme, tako i proizvođača višeg, softverskog i protokolskog sloja iste. Postojanje uređaja različitih proizvođača zahteva njihovu interoperabilnost. Samim tim se poznavanje protokola koji nisu vezani za konkretnog proizvođača pokazuje kao bitno u karijeri inženjera ove struke.

Takođe, u današnje vreme se javlja tendencija ka svojevrsnom odvajanju od fizičkog sloja tj. apstrakcijom hardvera različitim tehnikama virtuelizacije. Tehnike virtuelizacije, osim za virtuelizaciju hardvera operativnih sistema, sve više nalaze primenu i u razvoju i nadgledanju mrežne infrastrukture, kao i infrastrukture za skladištenje podataka.

Osim infrastrukturne i konkretne primene, virtuelizacija nalazi značajnu primenu u simulaciji različitih sistema. Emulacijom različite opreme u mreži mogu se razmatrati specifični scenariji do kojih može doći, ali zbog osetljivosti servisa i potrebe za kontinuiranim radom ne možemo testirati u realnosti.

Iz navedenih razloga, u pogledu nezavisnosti tehnologija, u ovom radu će biti prikazana konfiguracija OSPF (*Open Shortest Path First*) protokola koji je razvijen od strane međunarodne zajednice IETF (*Internet Engineering Task Force*). Takođe, tehnikom virtuelizacije, pomoću besplatnih alata *Virtualbox* i GNS3 simulatora, biće kreirana virtuelna sredina za testiranje ovog protokola. Aspekt kontinuiranog funkcionisanja mrežnih servisa biće razmatran prikazom restauracije mreže nakon njenog prekida pomoću OSPF protokola. Takođe, pošto će biti prikazana i implementacija BFD (*Bidirectional Forward Detection*) protokola u okviru OSPF-a, koji je protokol koji je nastao kao koncept od strane kompanije *Juniper*, test ove topologije biće prikazan na virtuelnim ruterima upravo ove kompanije.

2. PROTOKOLI RUTIRANJA

Pouzdan i efikasan prenos informacija i podataka, odnosno njihovo rutiranje je od presudnog značaja u današnjem društvu i direktno uslovljava njegov ekonomski prosperitet. Rutiranje predstavlja glavni proces kojim se upravo omogućava prenos informacija sa jedne na drugu lokaciju.

Rutiranje se obavlja pomoću tabele rutiranja, koja predstavlja bazu podataka o dostupnosti pojedinih mreža, i po načinu na koji se ona popunjava razlikujemo statičko i dinamičko rutiranje. Dinamičko rutiranje obavljaju protokoli rutiranja i omogućava ažurnost tabele rutiranja.

2.1. Klasifikacija internih protokola rutiranja

U zavisnosti od toga da li se rutiranje vrši unutar ili između dva autonomna sistema, razlikujemo interne (*IGP – Interior Gateway Protocols*) i eksterne (*EGP – Exterior Gateway Protocols*) protokole rutiranja. Osim ove podele, možemo napraviti podelu po načinu rada, pa tako razlikujemo:

- Rutiranje vektorom razdaljine (*distance vector*). Ovi protokoli mere broj „skokova“ do udaljene lokacije, sa maksimumom 16, i biraju putanju sa najmanjim brojem skokova. Najpoznatiji protokoli ovog tipa su RIP i EIGRP.
- Protokole stanja linka (*link state*), koji dodeljuju svakom interfejsu faktor cene, na osnovu čega zajedno sa stanjem datog interfejsa formiraju tabelu rutiranja. Ovi protokoli koriste mrežne informacije od svih rutera u datoj mreži, dok *distance vector* protokoli koriste samo informacije od rutera sa kojima su direktno povezani. Najpoznatiji protokoli ovog tipa su OSPF i IS-IS.
- Protokole vektora putanja (*path vector*), koji čuvaju informacije o kompletnim putanjama do odredišta.

3. VISOKA RASPOLOŽIVOST I KONVERGENCIJA MREŽE

Zbog sve veće zastupljenosti prenosa podataka i vremenski osetljivih servisa u sve više aspekata svakodnevnog čovekovog funkcionisanja, uvodi se pojam visoke raspoloživosti. Ovaj koncept se koristi kako u pogledu različitih servisa u mreži, tako i same mreže. Visoka raspoloživost koja uslovljava njenu brzu konvergenciju tj. detekciju njenog prekida i oporavak, u današnje vreme je od presudnog značaja.

Visoka raspoloživost se postiže planiranjem alternativnih putanja, gde nakon prekida jedne putanje, svi servisi mogu da nastave da funkcionišu. Da bi servisi imali što bolje performanse, u slučaju prekida nekog linka, potrebno je obezbediti brzu mrežnu konvegenciju.

Konvergencija mreže je pojam koji se može koristiti višestruko. U ovom radu ćemo pod terminom konvergencije mreže podrazumevati da tabele rutiranja ostaju nepromenjene određen vremenski period. Taj period može biti definisan prema vremenskom maksimumu koji je očekivan za stabilizaciju date mrežne topologije.

Mrežna konvergencija u mrežama sa standarnim IGP (*Interior Gateway Protocol*) mehanizmima može se nazvati i mrežna restauracija, s obzirom da se postojeće konekcije obnavljaju. Problem spore konvergencije najbolje prevazilaze protokoli stanja linka, koji između ostalog imaju kompletniji i ažurniji prikaz celokupne mreže.

4.OSPF

U klasu *link-state* protokola rutiranja spada OSPF (*Open Shortest Path First*), koji je najrasprostranjeniji interni protokol rutiranja. Za razliku od EIGRP-a, takođe jako rasprostranjenog protokola razvijenog od strane kompanije *Cisco* i funkcioniše samo na njihovim proizvodima, OSPF nije vezan za određenog proizvođača. Zasniva se na Dijkstra algoritmu za proračunavanje najjeftinijih putanja. Zamišljen je kao naslednik RIP protokola, a blizak je IS-IS protokolu.

Kada se koristi OSPF, ruter proverava stanje linkova i njihovu cenu, koja se dodeljuje interfejsu. Ruta se postavlja preko najjeftinije putanje, tj. putanje sa najmanjom cenom. Kada se veza uspostavi ili prekine, generiše se objava o stanju veze (*link-state advertisement* - LSA). Svaki ruter na kome je konfigurisan ovaj protokol čuva posebno strukturiranu bazu podataka koja se označava kao *Link-State Database* (u daljem tekstu LSDB). Svaka ruta u mreži je predstavljena jednim unosom u ovu bazu podataka i svaka ima pridruženu metriku.

Protokol postoji u nekoliko verzija: OSPFv2 i OSPFv3. OSPFv2 je starija verzija protokola namenjena za rad u IPv4 okruženju, a OSPFv3 je verzija protokola za IPv6 mreže. U ovom radu će se koristiti OSPFv2 verzija.

Kako bi se obezbedila podrška za velike mreže, OSPF uvodi podelu na oblasti (*Areas*). Primarna motivacija za uvođenje OSPF oblasti bio je problem održavanja LSDB baza aktuelnim, kao i opterećenje pri računanju SPF algoritma koje je značajno u slučaju velikog broja rutera u topologiji mreže. Ove oblasti su numerisane celim brojevima i međusobno povezane, ali nezavisne. Osnovna oblast je okosnica mreže, tzv. *Backbone* oblast na koju su povezane ostale oblasti. Ukoliko neka oblast ne može biti povezana direktno na *Backbone* oblast, potrebno je stvoriti virtuelni link ka njoj.

U odnosu na to kojoj oblasti pripadaju, iz perspektive OSPF protokola razlikujemo četiri tipa rutera:

- *Internal Routers* (IR)

Ovo su interni ruteri, međusobno povezani sa drugim ruterima i mrežama u okviru iste oblasti i nijednim od svojih interfejsa ne pripadaju drugim oblastima. Samim tim čuvaju LSDB samo od svoje oblasti, tj. imaju informacije samo o stanju svoje oblasti.

- *Backbone Routers* (BR)

Ovo su ruteri osnovne *Backbone* oblasti.

- *Area Border Routers* (ABR)

Ruteri koji pripadaju više oblasti, oni čuvaju LSDB od svih oblasti kojima pripadaju.

- *Autonomous System Boundary Router* (ASBR)

Ruteri koji se nalaze na granici oblasti u mreži unutar koje je aktivan OSPF protokol. Oni paralelno, pored OSPF-a koriste još neki dinamički protokol rutiranja ili imaju aktivne statičke rute ka drugim mrežama.

5. FAST HELLO MEHANIZMI I BFD

5.1. Koncept *fast-hello* mehanizama

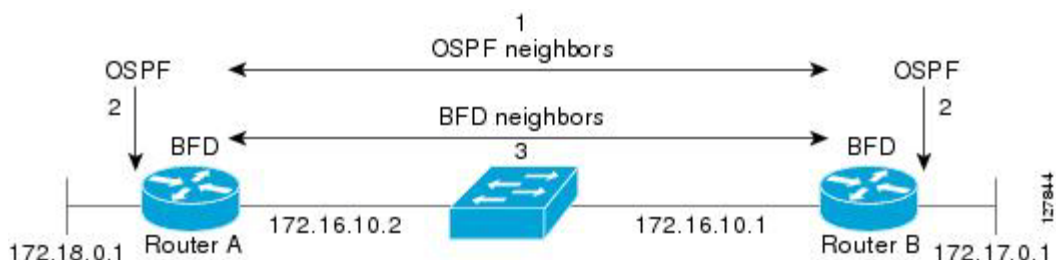
Detekcija pada linka u intervalima kraćim od sekunde je podržana u mnogim mrežnim protokolima ali i u OSPF-u. Protokoli OSPF i ISIS podržavaju *fast hello* pakete, *dead* intervale od jedne sekunde i *hello* pakete kraće od jedne sekunde. Problem ovih mehanizama je u tome što značajno opterećuju rad procesora rutera, pa u mrežama u kojima ruteri imaju puno suseda ovo značajno utiče na njihove performanse. Idealnu alternativu u ovim slučajevima pruža protokol BFD (*Bidirectional Forward Detection*), koji se isto zasniva na korišćenju intervala kraćeg od jedne sekunde, koji mogu biti implementirani na modulima interfejsa, čuvajući resurse procesora i kontrolne ravni.

5.2. BFD

BFD protokol konfiguriramo u cilju bolje konvergencije mrežnog stanja i omogućava nam brzu detekciju pada linka između mrežnih elemenata koji nemaju mehanizame za detekciju, ili su prespori i neodgovarajući. Može se reći da njegovim konstantnim korišćenjem u uniformnim vremenskim intervalima, radije nego skraćivanjem *hello* intervala različitih protokola, BFD postaje efikasan metod detekcije kvara u mreži, olakšava njeno planiranje i oporavak. Ovaj protokol je nezavisan od protokola rutiranja koji se koristi iznad njega. Naime, on nije protokol koji služi za kontrolu konektivnosti, već prosleđuje informacije o stanju linka hijerarhijski višem protokolu, koji nadalje obrađuje te informacije. BFD se može konfigurisati uporedo sa svim često korišćenim WAN i LAN protokolima rutiranja, kao što su OSPF, BGP, VRRP kao i uporedo sa statičkim rutiranjem.

BFD protokol se mora konfigurisati na svim ruterima u mreži i zahteva uspostavljanje BFD sesije. Uspostavljanje i raskidanje BFD sesije se vrši po principu *three-way handshake*-a, čime se omogućava da su oba učesnika u komunikaciji svesni stanja međusobne veze. Ruteri na kojima je uspostavljena BFD sesija nazivaju se *peer*-ovi.

Osim što se treba definisati za svaki ruter, BFD protokol neće funkcionisati sve dok se ne primeni na samu konfiguraciju protokola rutiranja na željenom interfejsu rutera.



Slika 5.1.1. Implementacija BFD protokola u okviru OSPF protokola

BFD paketi su enkapsulirani u UDP (*User Datagram Protocol*) paketima. BFD ima dva tipa paketa, *echo* i *control* pakete.

Originalna verzija je imala opciju samo za kontrolne pakete, koji se procesiraju od strane procesora. Novijom verzijom iz 2010. godine uvode se *echo* paketi čijom cirkulacijom se ne opterećuje kontrolna logička ravan, već komunikacija ostaje na hijerarhijski nižoj, prosleđivačkoj ravni. Naravno, postoji opcija isključivanja razmene *echo* paketa, vezujući komunikaciju za kontrolnu ravan.

Pomoću *control* paketa se uspostavlja inicijalna BFD sesija. Po prijemu kontrolnog paketa BFD process koji se izvršava na ruteru proverava tajmere i odgovara isto kontrolnim paketom.

Razmena *echo* paketa omogućena je time što, pri slanju paketa se i za odredišnu i izvorišnu IP adresu postavlja upravo adresa tog interfejsa rutera. Dakle, po prijemu paketa, mehanizmi prosleđivanja samo gledaju postojeću tabelu suseda i na osnovu nje prosleđuju *echo* pakete. Kada je uključen *echo* režim rada, tada je upravo gubitak kratkih *echo* paketa taj koji je odgovoran za alarmiranje rutera o padu određenog linka i obezbeđuje nam oporavak mreže u kratkim vremenskim intervalima.

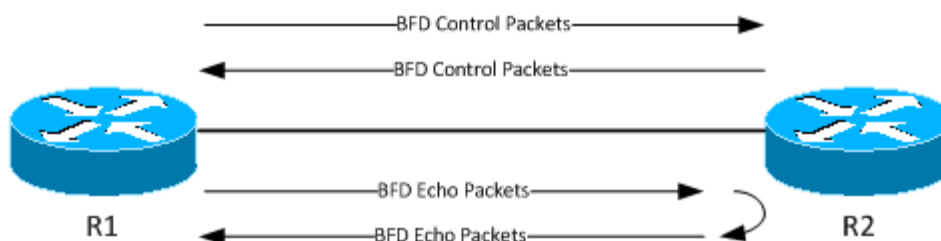


Figure 1. BFD Echo Function

Slika 5.1.2. Prikaz razmene *hello* i *echo* paketa u okviru jedne BFD sesije

BFD uvek funkcioniše u *unicast*, *point-to-point* režimu. Između dva BFD *peer*-a može biti uspostavljeno više BFD sesija, ako u jednom od dva pravca postoji višestruke putanje, iako možda na drugoj postoji jedna ili manje njih. Svaki sistem procenjuje koliko brzo može da prima i šalje pakete, na osnovu čega sa susedima može da se dogovori o tome koliko brzo može da detektuje prekid. Ove vrednosti se uzimaju kao vrednosti parametara, koji mogu biti menjani u realnom vremenu, omogućavajući prolagođavanje potencijalnim promenama u mreži. Ovim se takođe omogućava da učesnici u komunikaciji koji imaju veću procesorsku moć da detektuju prekid brzo, ali i da dozvole “sporijim” učesnicima da učestvuju.

BFD je operativan u dva režima, asinhronom i *on-demand*.

Asinhron režim rada je primarni režim. Po ovom režimu, sistem periodično šalje pakete, i ako ih ne stigne uzastopce određen broj, komunikacija se smatra prekinutom.

Drugi režim je *on-demand* režim. U ovom režimu se podrazumeva da sistem unutar koga je implementiran BFD ima nezavisan način utvrđivanja povezanosti sa drugim ruterima. Nakon uspostavljanja sesije, prestaje razmena komunikacije kontrolnim paketama, osim periodično, ali kada se eksplicitno traži potvrda postojanja komunikacije. Tada se razmeni određen broj kontrolnih paketa, nakon čega se komunikacija utišava.

Implementacija ovog protokola se razlikuje na ruterima različitih proizvođača, tj. njihovim operativnim sistemima.

5.2.1. Konfiguracija na Cisco uređajima

Na ruterima proizvođača *Cisco* zahteva se da CEF (*Cisco Express Forwarding*) bude aktivan.

BFD je potrebno konfigurisati na svim ruterima za koje želimo da bude aktivan BFD protokol, i to činimo zadavanjem komande:

bfd interval vrednost min_rx vrednost multiplier vrednost.

Parametar *interval* određuje razmenu *echo* tj. iskazuje željenu brzinu razmene *echo* paketa. Parametar *min_rx* označava minimalno vreme za koje možemo da obradimo pakete, i osigurava da, iako drugi interfejs možda ima kraći *interval*, ne može da nam šalje brže nego što mi možemo da primimo *echo* paket. Parametar *multiplier* je ceo broj, i njegova uloga je takva da ako se *echo* paket ne vrati za celobrojni umnožak koji predstavlja parametar *interval*, proglašavamo *peer*-a mrtvog.

Kao što je već naglašeno, BFD neće funkionisati osim ako se ne aktivira unutar određenog protokola. Kada ruter detektuje gubitak BFD paketa, on obaveštava protokol rutiranja unutar koga je definisan, čime se nadalje vrši ažuriranje tabela rutiranja.

Aktiviranje BFD protokola vršimo na pojedinačnim interfejsima na kojima je podignut OSPF komandom:

bfd all-interfaces

5.2.2. Konfiguracija na Juniper mrežnim uredajima

Na ruterima kompanije *Juniper*, u njihovom operativnom sistemu *Junos*, konfiguracija ovog protokola se razlikuje od konfiguracije na ruterima kompanije *Cisco*. Nažalost, još uvek ne postoji opcija za *echo* režim rada, čime se razmena paketa koji testiraju mrežu i dalje vrši na kontrolnoj ravni.

Bez obzira na to, i dalje se implemenacijom ovog protokola unutar OSPF-a, postižu bolje performanse u pogledu mrežne konvergencije i u odnosu na *fast-hello* mehanizme manje se troše procesorski resursi imajući u vidu zaglavlja paketa koji se razmenjuju.

Parametri koji se podešavaju u okviru operativnog sistema *Junos* su:

- **detection-time threshold**

Ovaj parametar predstavlja marginu adaptacije sa zadatim vremenom detekcije. Kada se BFD sesija adaptira na zadato vreme detekcije, okida se *trap* signal i pravi se log o tome.

- **full-neighbors-only**

Ovim parametrom je moguće odrediti da se BFD sesija uspostavlja u okviru OSPF-a samo ako su u stanju *full-adjacency*.

- **minimum-interval**

Ovim parametrom podešavamo i minimalni interval u kojem ruter šalje *hello* pakete i minimalni vremenski period u kome ruter očekuje da primi odgovor od suseda. Ovi intervali su reda veličine milisekundi. Takođe je moguće ove parametre konfigurisati zasebno kao:

- **transmit-interval minimum-interval**

Minimalni vremenski interval u kome ruter može da šalje *hello* pakete.

- **minimum-receive-interval**

Minimalni interval detekcije kvara na linku. Određuje vremenski interval u kome ruter očekuje da primi *hello* paket od rutera sa kojim je u BFD susedstvu.

- **multiplier**

Ovim parametrom definišemo nakon koliko paketa će se BFD sesija smatrati prekinutom.

- **no-adaptation**

Ovom komandom možemo da ne dozvolimo BFD sesiji da se adaptira na promene u mreži.

- **transmit-interval threshold**

Ovaj parametar predstavlja marginu adaptacije vremena slanja *hello* paketa, koja mora biti veća od minimalnog vremena slanja. Ako se za vreme slanja pređe ova margina pri adaptaciji, šalje se *trap* signal i upisuje log poruka.

- **version**

Verzija BFD protokola se automatski podešava, ali je moguće podesiti je ovom komandom.

Da bi konfigurisali BFD protokol u okviru OSPF-a, potrebno je da uspostavimo sesije na interfejsima koji učestvuju u komunikaciji. Ovo činimo komandom **bfd-liveness-detection** koju

nadovezujemo na konfiguraciju interfejsa u okviru konfiguracije OSPF protokola. Na ovo nadovezujemo i podešavanje pojedinačnih parametara ovog protokola.

Kao primer, u okviru oblasti 0 na interfejsu *fe-0/0/1* podešavanje minimalnog vremena detekcije na 300 milisekundi se može obaviti na sledeći način:

```
set protocols ospf area 0.0.0.0 interface fe-0/0/1 bfd-liveness-detection minimum-interval 300
```

```
set protocols ospf area 0.0.0.0 interface fe-0/0/1 bfd-liveness-detection multiplier 4
```

```
set protocols ospf area 0.0.0.0 interface fe-0/0/1 bfd-liveness-detection full-neighbors-only
```

Ovime smo podesili minimalno vreme detekcije i transmisije na 300 milisekundi i uslov da se BFD sesija uspostavi samo u slučaju kada je ostvaren *full-adjacency*. Takođe, nakon isteka četiri minimalna vremenska intervala detekcije, BFD sesija će biti prekinuta. Postojeće BFD sesije možemo videti iz operacionog režima rada rutera zadavanjem komande:

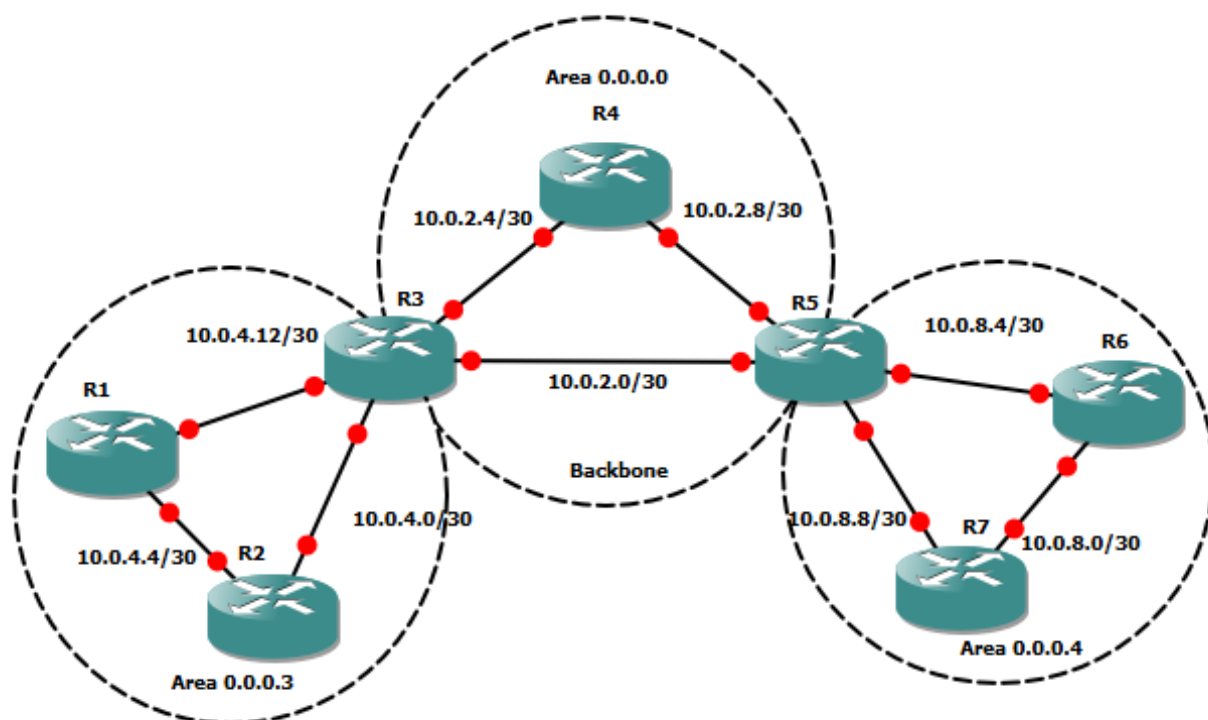
```
show bfd session
```

6. TOPOLOGIJA TEST MREŽE

Izabrana topologija u ovom radu je izabrana na taj način da odgovara uobičajenim primenama u praksi. Mrežna topologija koju planiramo da testiramo u okviru ovog rada prikazana je na slici 6.1. Mrežu smo podelili na tri logičke oblasti u pogledu OSPF protokola, koje se u ovoj terminologiji označavaju kao zone ili oblasti (*Area*). Zona 0 je takozvana *Backbone* oblast i povezana je na oblasti tri i četiri preko ABR (*Area Border Router*) rutera R3 i R5. Zone su na slikama i u konfiguraciji označavane u formatu koji definiše RFC, npr. zona 3 je 0.0.0.3. U pogledu raspoloživosti mreže, postoji više scenarija u zavisnosti od toga na kom delu mreže je kvar. Razlikujemo tri scenarija:

- 1) U slučaju prekida jednog od linkova ili pada pojedinačnih interfejsa postoji redundansa, tj. svi servisi će neometano nastavljati da funkcionišu u ovom slučaju.
- 2) Prestankom rada jednog od rutera R1, R2, R4, R6 i R7, servisi nastavljaju neometano da funkcionišu.
- 3) Prestankom rada jednog od ABR (*Area Border Router*), R3 ili R5, ruteri u pojedinačnim oblastima mreže će nastaviti da komuniciraju na nivou oblasti, ali će komunikacija između ovih oblasti biti u prekidu.

Nadalje u radu, nakon prikaza podešavanja konfiguracije mreže, biće prikazani svi navedeni scenariji.



Slika 6.1. Prikaz topologije test mreže

6.1. Simulacija testne mreže

U ovom radu će biti testiran rad OSPF protokola i njegova sposobnost oporavka mreže na ruterima kompanije Juniper. Biće korišćeni besplatni alati otvorenog koda, simulator *GNS3* i hipervizor *Virtualbox*.

Detaljne konfiguracije svakog rutera su date u prilogu **A**.

Adrese interfejsa i njihova raspodela po OSPF oblastima su dati u prilogu **B**.

6.1.1. *GNS3 Simulator*

GNS3 je softverski paket nastao 2008. godine koji objedinjuje skup programa za emulaciju, virtuelizaciju i simulaciju mrežnih uređaja.

Softverska emulacija je izvršavanje softvera jednog uređaja u program na drugom uređaju (u emulatoru). Emulatori u sebi sadrže predstavu hardvera uređaja čiji softver izvršavaju, i izvršavajući softver oni menjaju stanje predstave hardvera na isti način kao što bi se menjalo u pravom uređaju.

Virtuelizacija predstavlja razdvajanje softvera i hardvera i podrazumeva izvršavanje jednog operativnog sistema u okviru drugog operativnog sistema. Virtuelizacija podrazumeva “podizanje” virtuelnog uređaja (ili čitave mreže) na računaru opšte namene. Virtuelizovani sistemi mogu ostvariti izuzetnu vernost, čak bolju nego kod hardverskih modela, uz veliku skalabilnost; takođe je moguće povezati virtuelni uređaj sa stvarnim, pa tako ostvariti jedan vid emulacije. Mana virtuelizacije je što ovakvi sistemi rade sporije od realnog vremena, zbog ograničenosti procesorke moći računara na kojima se nalaze.

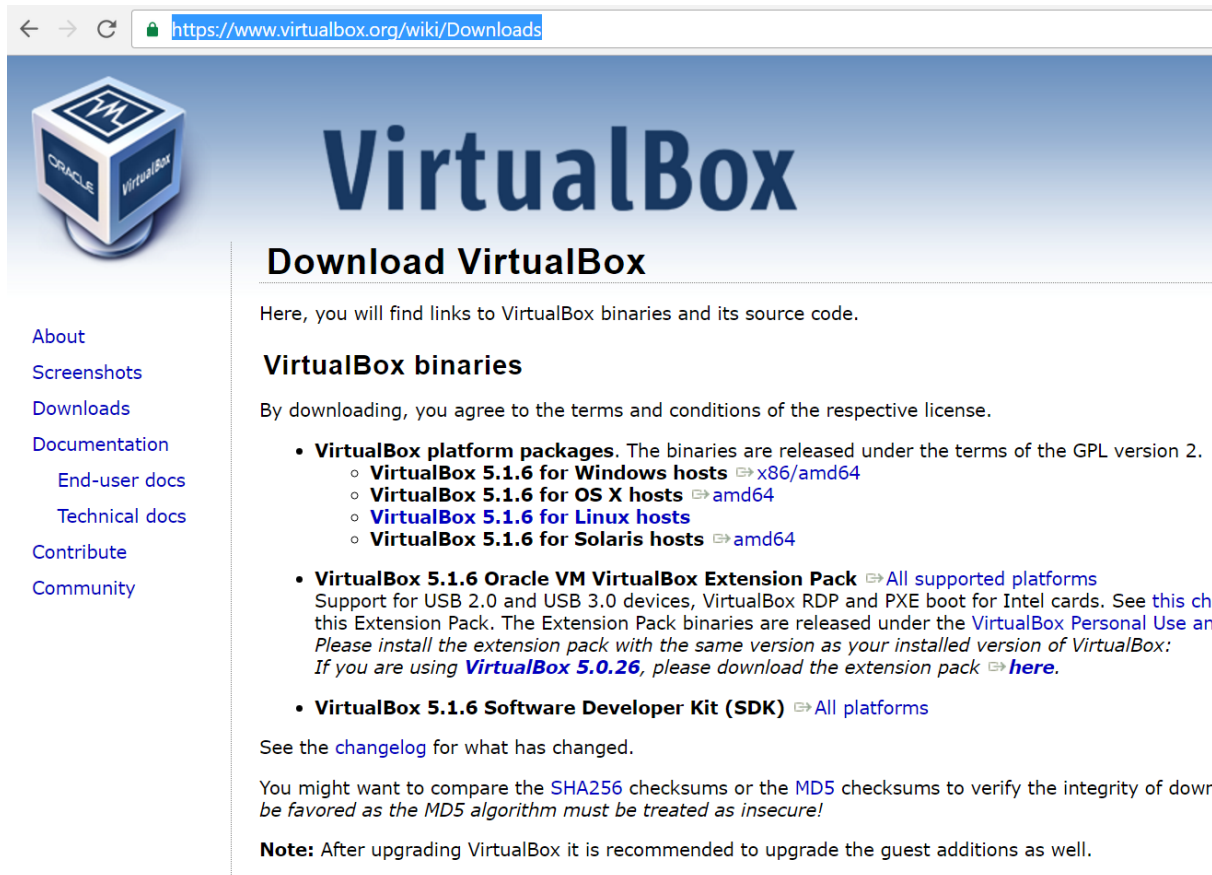
Programi koji emuliraju hardver nazivaju se hipervizori. Instance različitih operativnih sistema mozemo instalirati na hipervizor. Za emulaciju *Cisco* opreme GNS3 simulator koristi *Dynamips* hipervizor. Za emulaciju *Juniper* opreme, mogu se napraviti tzv. *Junos Oive* mašine, koje su u suštini virtualne mašine koje u osnovi imaju *FreeBSD* operativni sistem. GNS3 pruža intuitivni grafički interfejs (*GUI*) za dizajniranje i konfiguraciju virtuelnih mreža.

GNS3 nudi mogućnost simuliranja *Cisco* rutera, *Cisco PIX firewall-a*, *Cisco ASA firewall-a*, *Juniper* rutera, *ATM Bridge/Switch-a* i *Frame Relay Switch-a*. Iako su tu uvršćeni i *Cisco*-vi svičevi, simulacija ovih uređaja trenutno nije moguća. GNS3 je takođe usko integrisan sa *Wireshark*-om, programom za “hvatanje” paketa i analizu protokola. Iako je najviše fokusiran na podržavanje *Cisco* i *Juniper* opreme, moguće je emulirati hardver drugih vendora koristeći hipervizore poput *Qemu* i *VirtualBox* i *VMware*-ovih *VMware Fusion* i *Vmware Workstation*. Ovim nalazi primenu između ostalog i u simulaciji mreža sa SDN (*Software-Defined Networking*) opremom.

GNS3 simulator je dostupan kako za *Windows*, tako i za *Linux* i *MAC-OS* korisnike. Osim par razlika u koracima pri instalaciji, u samom softverskom paketu nema puno razlika. Može se preuzeti sa zvanične Internet stranice <https://www.gns3.com/>. Na zvaničnoj stranici pod odeljkom dokumentacije moguće je videti detalje i uputstva za instalaciju.

6.1.2. Alat Virtualbox

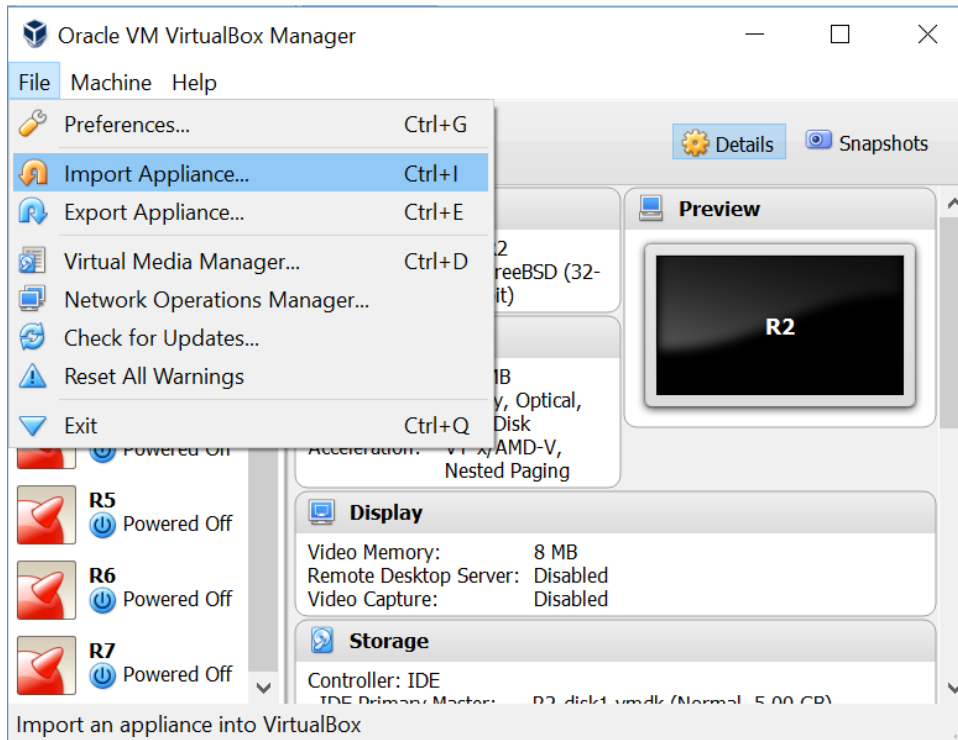
Alat *Virtualbox* može se preuzeti besplatno sa sajta <https://www.virtualbox.org/>. *Virtualbox* predstavlja hipervizor drugog nivoa, koji omogućava emulaciju jednog operativnog sistema u drugom.



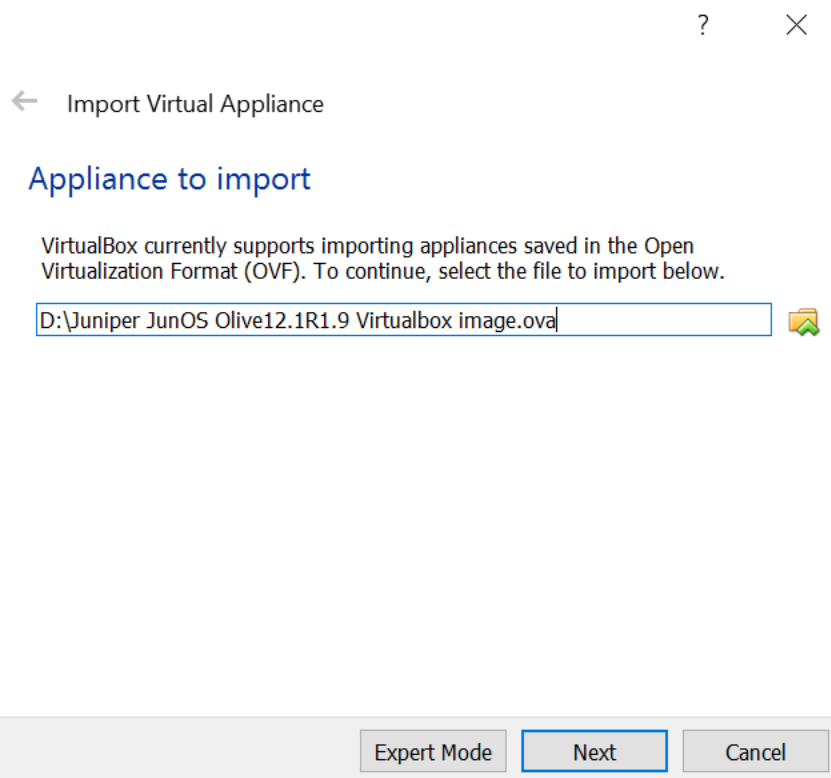
Slika 6.1.2.1. Veb stranica sa koje preuzimamo *Virtualbox* hipervizor

6.1.3. Kreiranje testne topologije

Nakon instaliranja ovog softvera i njegovog pokretanja, potrebno je kreirati virtuelnu mašinu sa *Junos* operativnim sistemom. Najlakši i najbrži način je instalacijom već prekonfigurisane virtuelne mašine u obliku aplajnsa (*appliance*) iz OVF (*Open Virtualization Format*) templejta, koja nosi ekstenziju *.ova*. OVF templejti predstavljaju format u koji se pakuje operativni sistem i omogućava bržu i lakšu instalaciju na virtuelnim mašinama. Importovanje aplajnsa prikazano je na slikama 6.1.2.2 i 6.1.2.3.

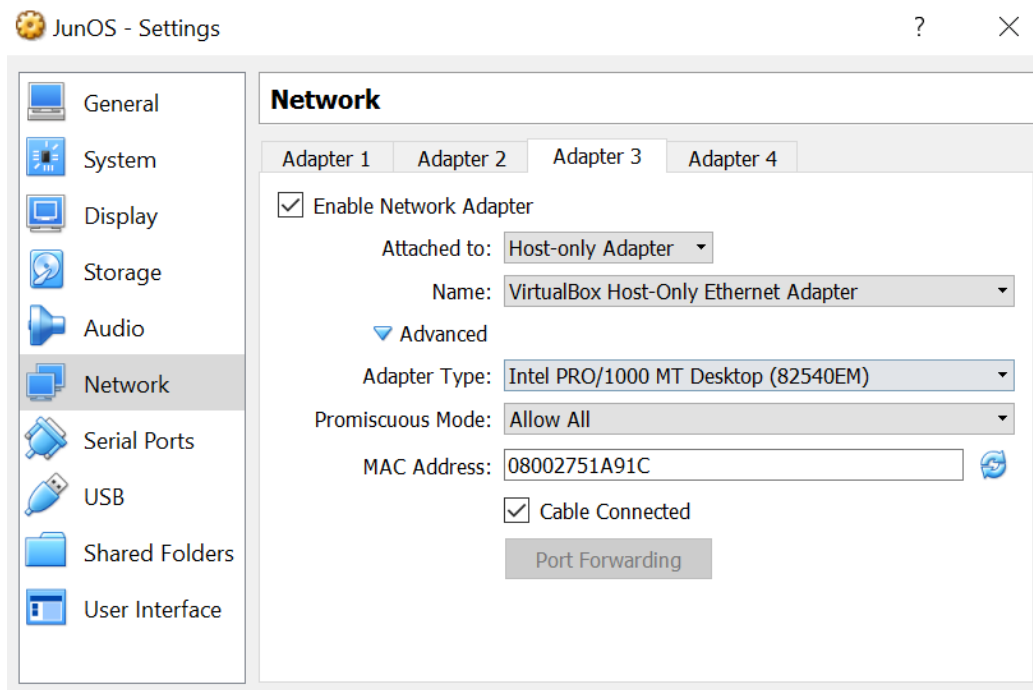


Slika 6.1.2.2. Importovanje operativnog sistema kao *appliance*



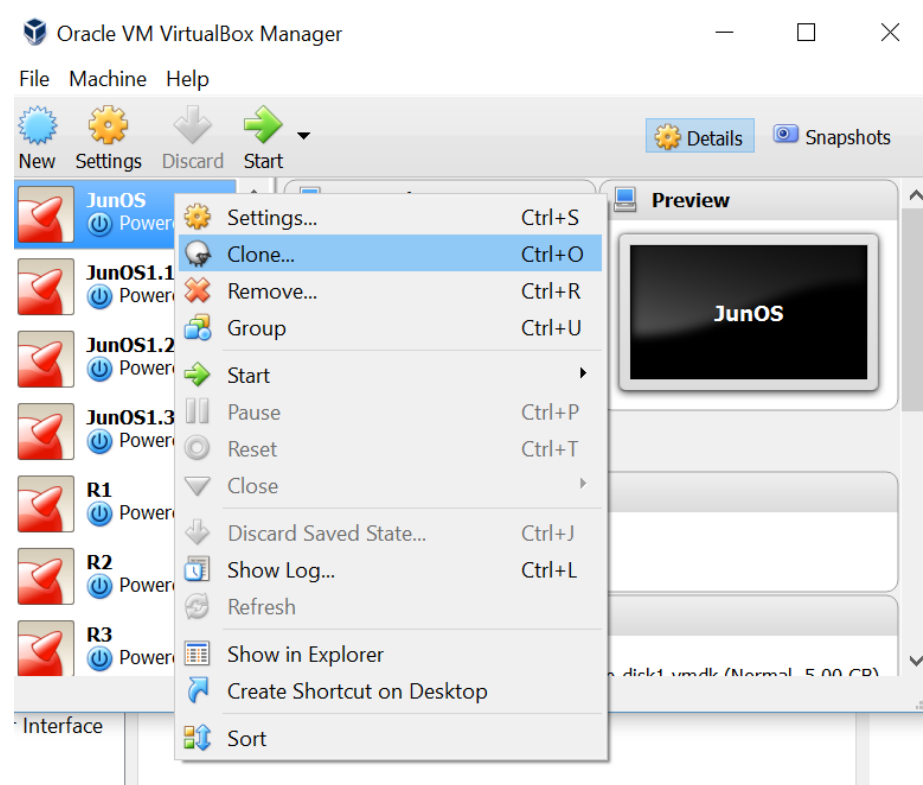
Slika 6.1.2.3. Importovanje operativnog sistema kao *appliance*

Verziju operativnog sistema *Junos* koju smo koristili za izradu ovog rada je 12.1R1.9. Nakon što smo napravili virtuelnu mašinu, da bi mogli da postignemo testiranje željene topologije, potrebno je da povećamo broj mrežnih adaptera na četiri, i da ih podesimo tako da nam je omogućena komunikacija između njih. Izabraćemo da mrežni adapteri budu *host-only*, čime ove virtuelne mašine stavljamo u isti sabnet.

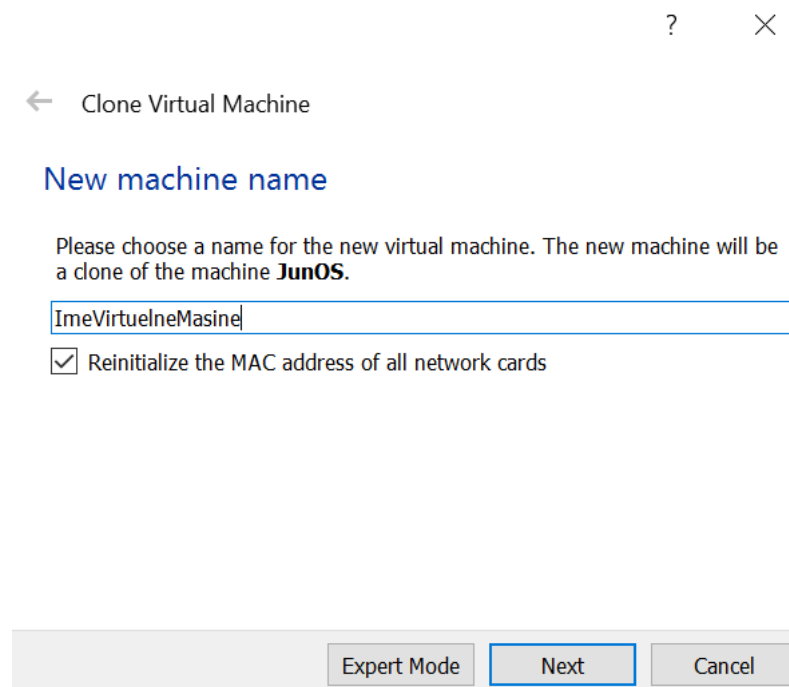


Slika 6.1.2.4. Podešavanje virtuelne mašine

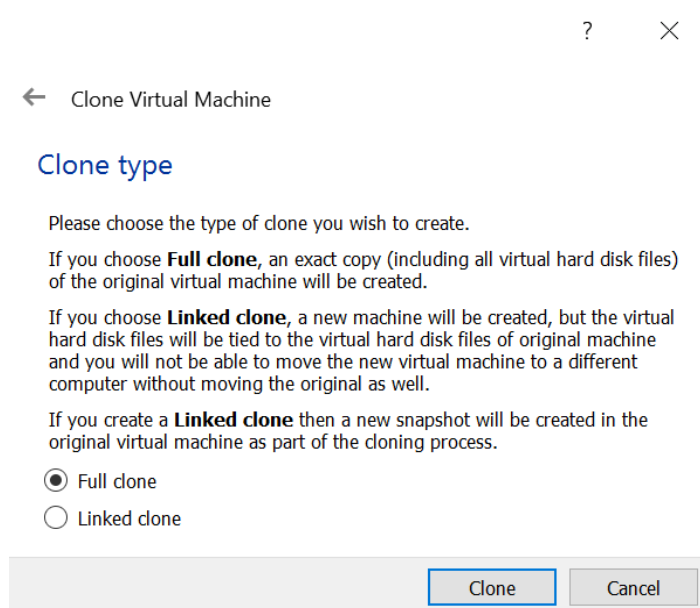
Pošto ćemo za potrebe rada konfigurisati sedam rutera, potrebno nam je dakle sedam virtuelnih mašina. S obzirom da su nam potrebne mašine iste konfiguracije, kloniraćemo postojeće. Ovo radimo izborom naše kreirane virtuelne mašine i biramo opciju Clone. Zadajemo ime mašine koju ćemo ovime krerati, i selektujemo opciju reinicijalizacije MAC adresa, jer nam je potrebno da interfejsi mašina u našoj topologiji imaju različite MAC adrese. Izabraćemo i opciju *Full clone*, ne bi li, kao što je već spomenuto, mašina imala identičnu konfiguraciju. Ovo možemo videti na slikama 6.1.2.5-7.



Slika 6.1.2.5. Prikaz opcije kloniranja virtualne mašine

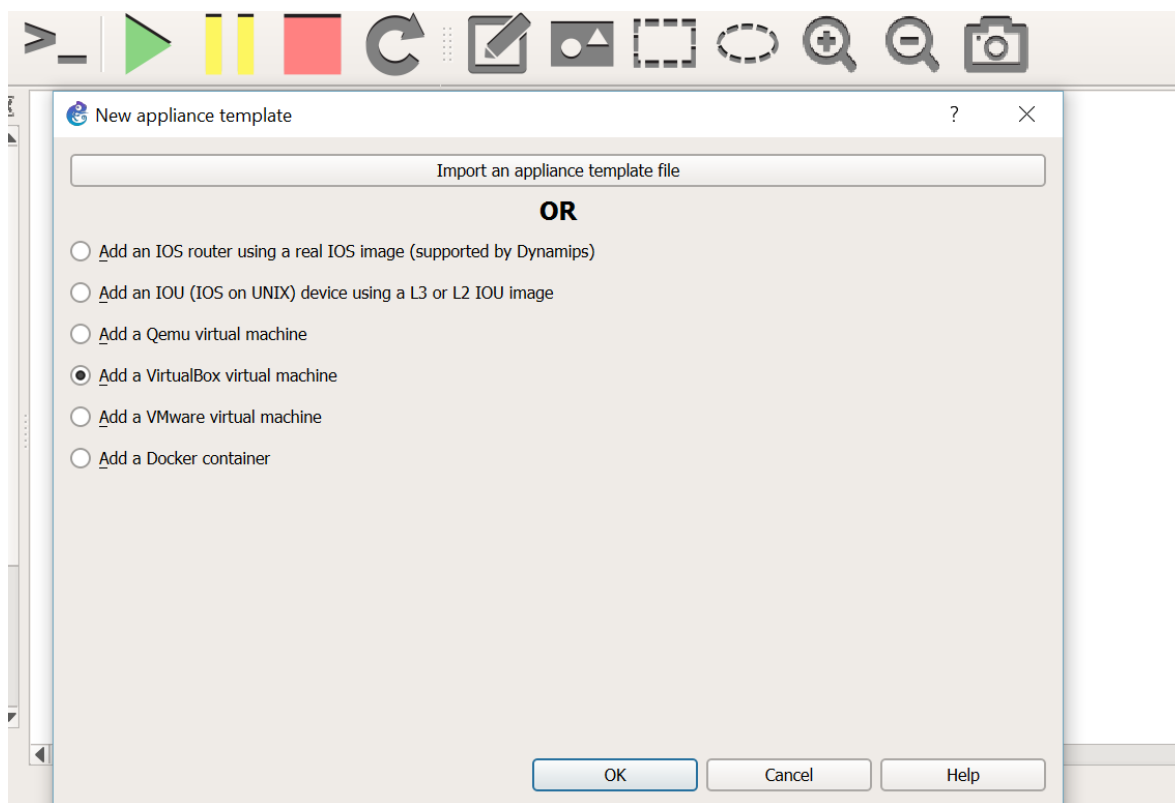


Slika 6.1.2.6. Prikaz opcije biranja imena klonirane virtualne mašine



Slika 6.1.2.7. Prikaz opcije biranja tipa klona virtuelne mašine

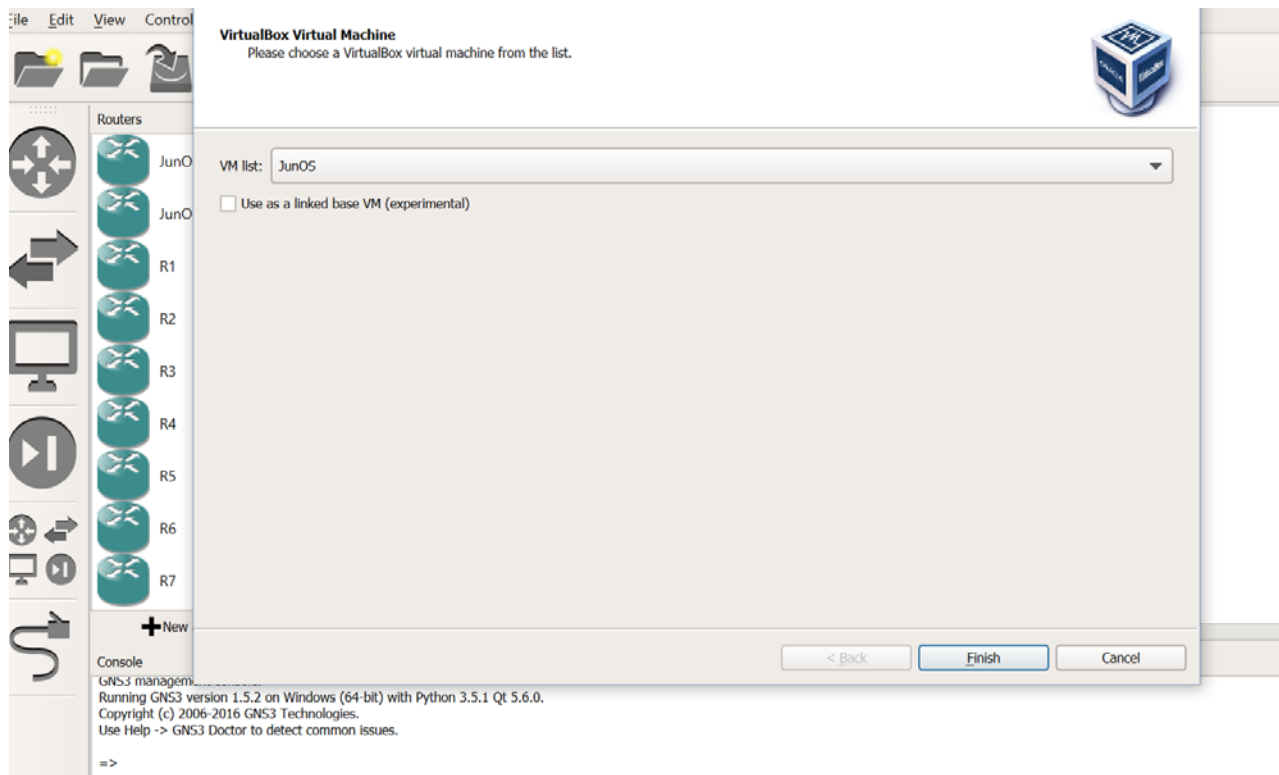
Nakon kreiranja potrebnih virtuelnih mašina, prelazimo na kreiranje test topologije u GNS3 simulatoru. Najpre treba da dodamo podignute virtuelne mašine u sam simulator. To radimo izborom *New Appliance Template*, kao na slici 6.1.2.8.



64-bit) with Python 3.5.1 Qt 5.6.0.
es.
on issues.

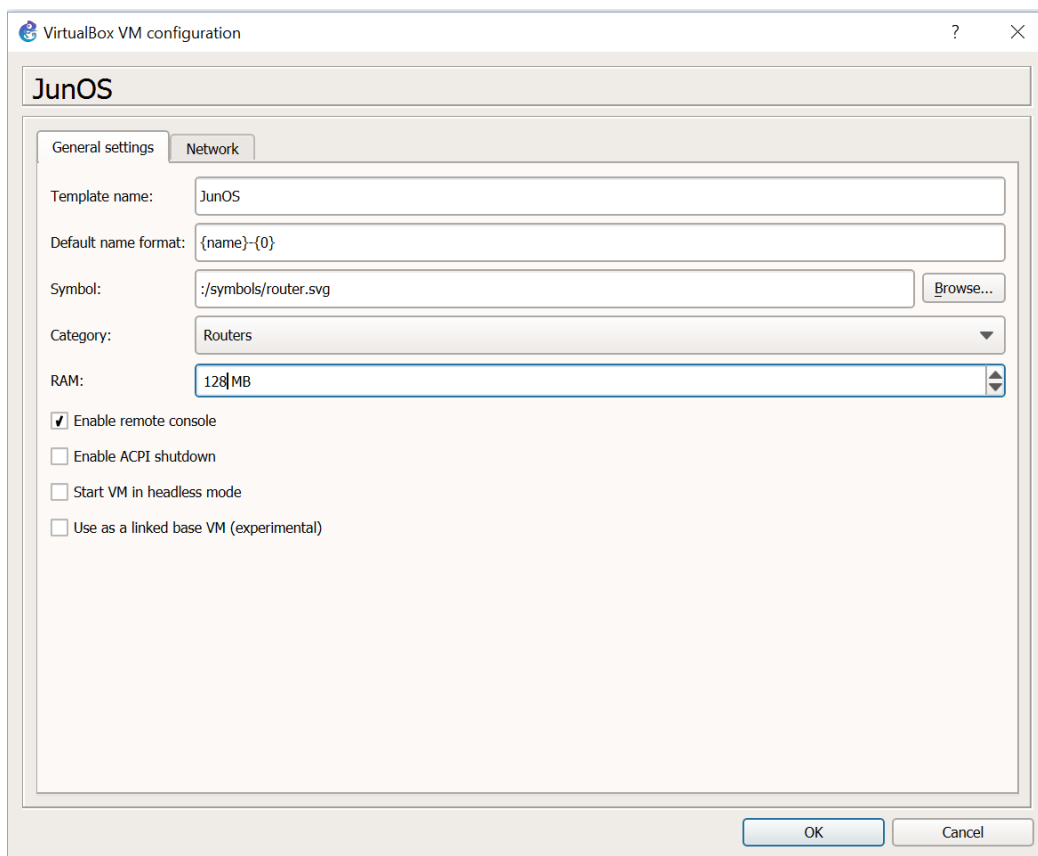
Slika 6.1.2.8. Dodavanje postojeće virtuelne mašine sa *VirtualBox* hipervizora

Nakon toga, iz ponuđene liste u padajućem meniju treba izabrati mašine koje želimo da dodamo, kao što se može videti na slici 6.1.2.9. i izabrati opciju *Finish*.



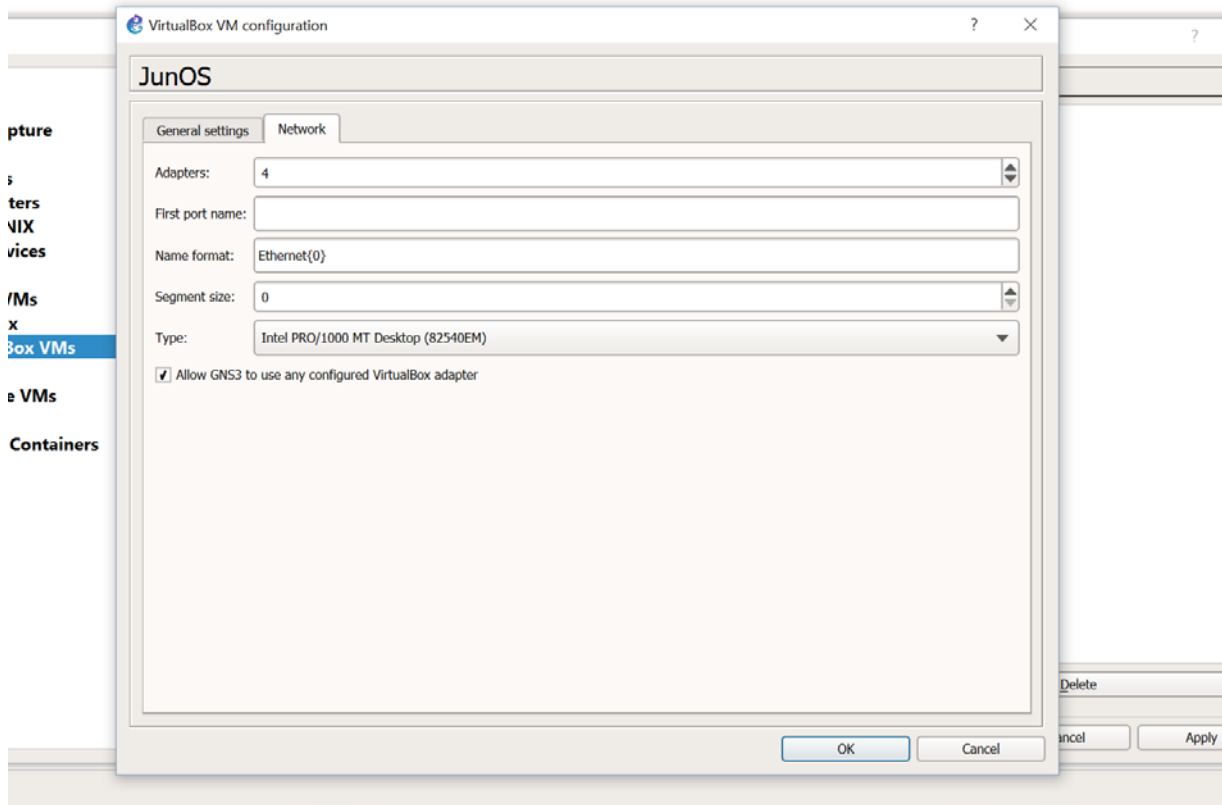
Slika 6.1.2.9. Izbor virtuelne mašine koju dodajemo u simulator

Nakon dodavanja ove mašine, potrebno je promeniti podrazumevana podešavanja. Podešavanja koja smo promenili su dozvoljavanje korišćenja konzole sa udaljene lokacije, količinu RAM memorije koju mašina može da koristi, kao i staviti je u odgovarajuću kategoriju sa odgovarajućim simbolom. Ovo je prikazano na slici 6.1.2.10.



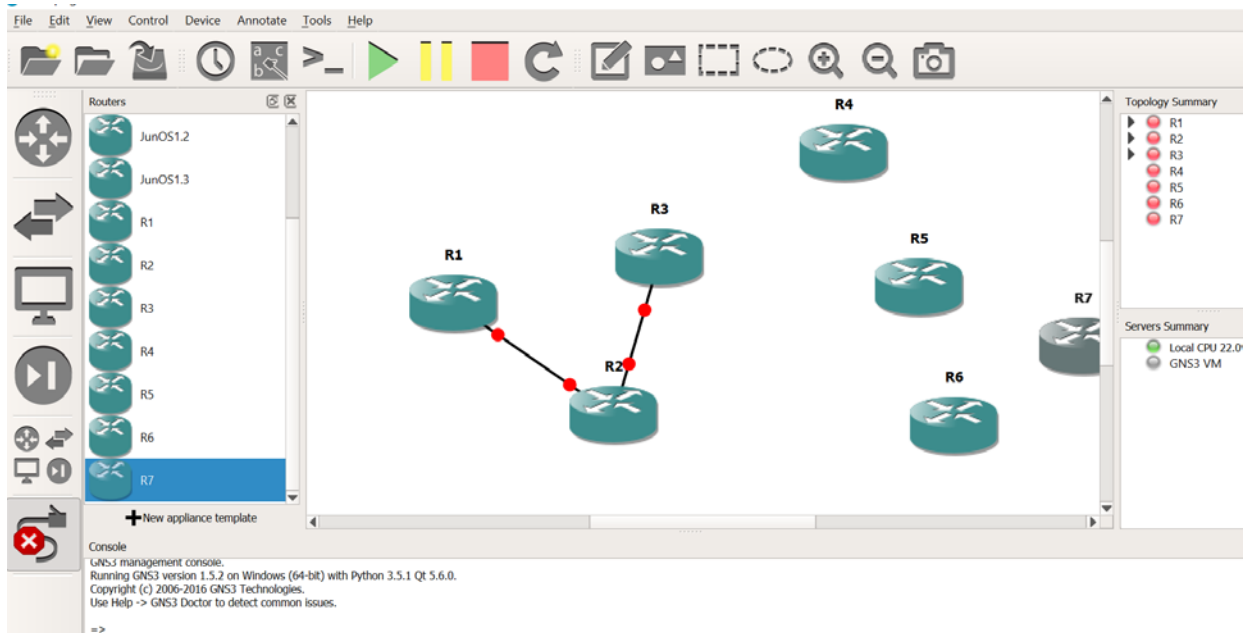
Slika 6.1.2.10. Podešavanja virtuelne mašine u okviru GNS3 simulatora

Takođe, potrebno je da podesimo broj ethernet adaptera na četiri i dozvoliti GNS3 simulatoru da koristi adaptore dodeljene hipervizorom Virtualbox. Ovo možemo da vidimo na slici 6.1.2.11.

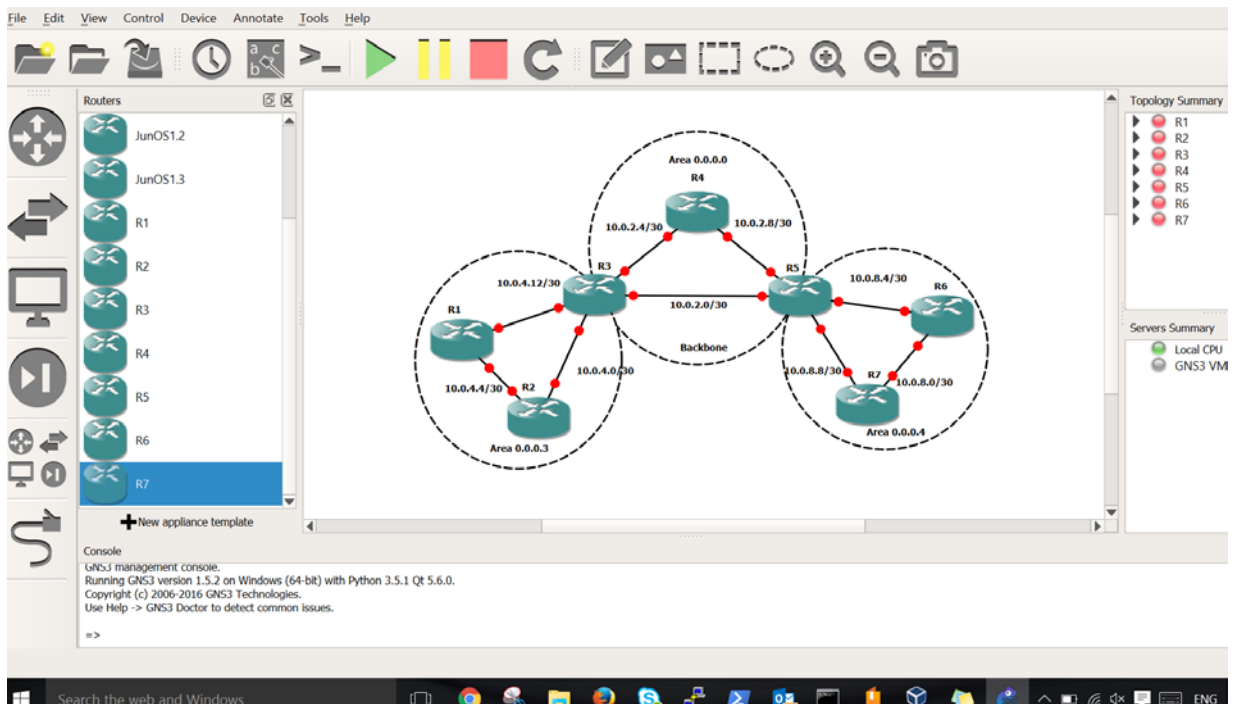


Slika 6.1.2.11. Podešavanja virtuelne mašine u okviru GNS3 simulatora

Nadalje ćemo dodati mašine u testnu topologiju i povezati ih. Na slici 6.1.2.12 prikazan je korak povezivanja rutera, a na slici 6.1.2.13 i krajnja topologija.



Slika 6.1.2.12. Prikaz povezivanja rutera u mreži



Slika 6.1.2.13. Prikaz kreirane krajnje testne topologije

6.2. Konfiguracija mrežnih uređaja

6.2.1. Konfiguracija interfejsa

Pri inicijalnom startovanju virtuelnog rutera, lozinka za nalog *root* nije podešena, tj. samo pritiskom tastera *enter*, možemo pristupiti interaktivnom interfejsu.

Da bi omogućili menjanje postojeće konfiguracije, u komandnoj liniji unosimo najpre komandu **cli**, a zatim **edit**. Prvi korak u procesu konfiguracije je adresiranje interfejsa rutera adresama IPv4 adresama. Za ruter R3, koji će komunicirati sa četiri interfejsa, to činimo zadavanjem sledećih komandi:

```
set interfaces em0 unit 0 family inet address 10.0.4.14/24
```

```
set interfaces em1 unit 0 family inet address 10.0.4.2/24
```

```
set interfaces em2 unit 0 family inet address 10.0.2.5/24
```

```
set interfaces em3 unit 0 family inet address 10.0.2.1/24
```

Da bi uspeali da sačuvamo konfiguraciju, potrebno je da podesimo lozinku. Ovo činimo zadavanjem komande:

```
set system root-authentication plain-text-password
```

Sada možemo sačuvati podešavanja zadavanjem komande **commit**.

6.2.2. Konfiguracija OSPF protokola

OSPF je protokol koji funkcioniše na pojedinačnim interfejsima rutera, pa ga tako moramo i podesiti. Za ABR ruter R3 i njegov interfejs *em2* za koji želimo da bude u zoni 0.0.0.0, unosimo komande:

```
set protocols ospf area 0.0.0.0 interface em2
```

Ovim metrika na ovoj putanji uzima podrazumevanu vrednost 1. Da bi, radi testiranja promene putanje u slučaju kvara, saobraćaj u inicijalnom stanju išao od zone 3 ka zoni 4 preko rutera R4, podesićemo metriku na interfejsu *em3* na 5. OSPF će birati putanju sa manjom metrikom. Dakle, na ruteru R3 konfiguraciju OSPF protokola obavljamo komandom:

```
set protocols ospf area 0.0.0.0 interface em3 metric 5
```

Nakon ovoga naredbom **commit** čuvamo unesene promene u konfiguraciji.

6.2.3. Konfiguracija BFD protokola

Konfiguracija BFD nije neophodna za funkcionisanje OSPF protokola. Ona može intenzivno opteretiti rad procesora. Ali, ako nam ovi resursi nisu problem, ovaj protokol donosi značajan benefit u pogledu detekcije pada linka. Konfiguriramo ga u okviru OSPF protokola, dakle vezujemo za konkretan interfejs, pa tako za interfejs *em0* rutera R3 konfiguracija se čini na sledeći način:

```
set protocols ospf area 0.0.0.3 interface em0 bfd-liveness-detection minimum-interval 300
```

```
set protocols ospf area 0.0.0.3 interface em0 bfd-liveness-detection multiplier 4
```

```
set protocols ospf area 0.0.0.3 interface em0 bfd-liveness-detection full-neighbors-only
```

Ovime smo podesili minimalno vreme detekcije i transmisije na 300 milisekundi i uslov da se BFD sesija uspostavi samo u slučaju kada je ostvaren *full-adjacency*. Takođe, nakon isteka 4 minimalna vremenska intervala detekcije, BFD sesija će biti prekinuta.

Aktivnost sesije možemo proveriti naredbom **show bfd session**. Na slici 6.2.3.1 prikazan je pregled aktivne sesije za ruter R1.

```
root> show bfd session
Address          State   Interface  Detect   Transmit
10.0.4.6         Up     em0.0      1.200   0.300   4
10.0.4.14        Up     em1.0      1.200   0.300   4

2 sessions, 2 clients
Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
root>
```

Slika 6.2.3.1. Prikaz pregleda aktivne bfd sesije rutera R1

Nakon konfigurisanja svih rutera u mreži, naredbom **show route** možemo videti aktivne putanje ka drugim ruterima u mreži, koje taj ruter vidi. Prikaz tabele rutiranja za ruter R3 dat je na slici 6.2.3.2.

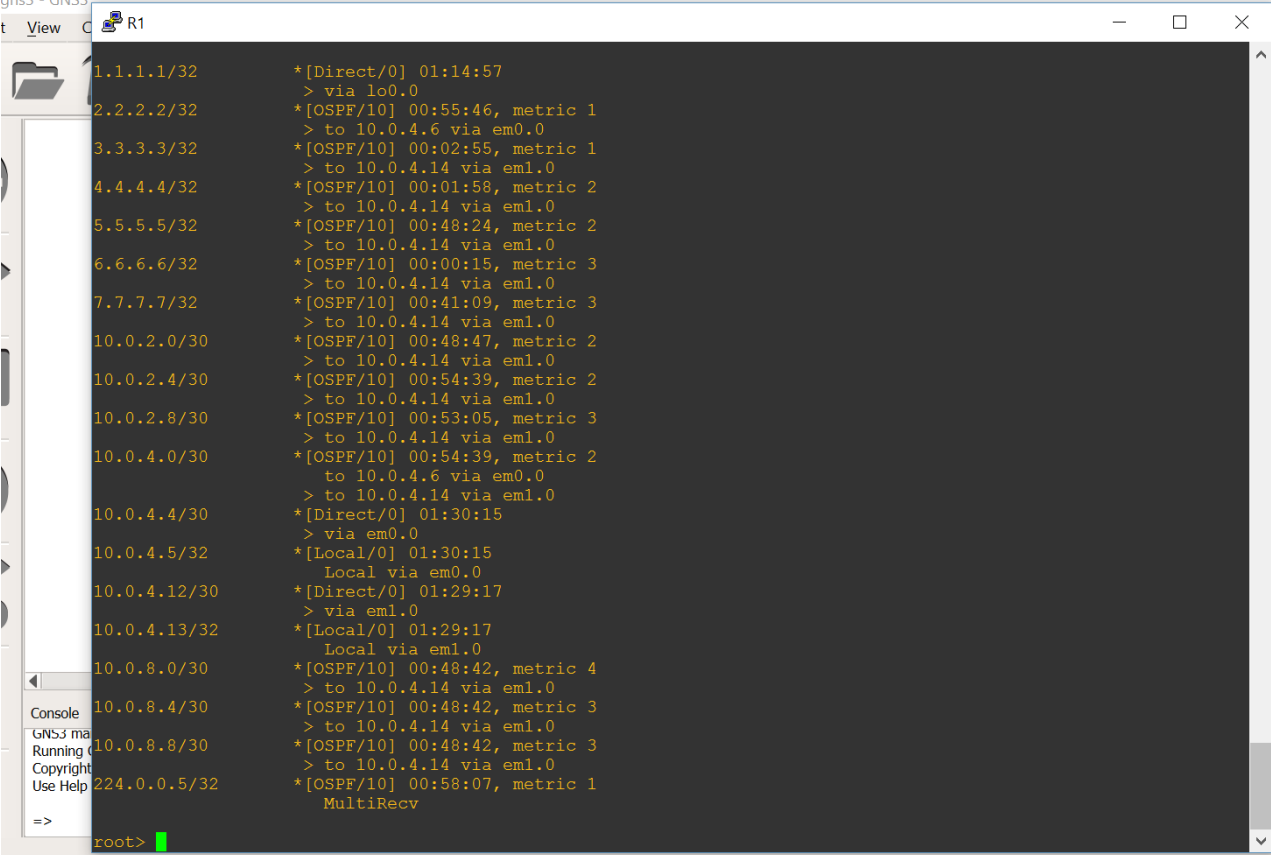

```
R3
1.1.1.1/32      * [OSPF/10] 01:15:51, metric 1
                 > to 10.0.4.13 via em0.0
2.2.2.2/32      * [OSPF/10] 01:15:51, metric 1
                 > to 10.0.4.1 via em1.0
3.3.3.3/32      * [Direct/0] 01:35:31
                 > via lo0.0
4.4.4.4/32      * [OSPF/10] 00:21:43, metric 1
                 > to 10.0.2.6 via em2.0
5.5.5.5/32      * [OSPF/10] 00:06:02, metric 1
                 > to 10.0.2.2 via em3.0
6.6.6.6/32      * [OSPF/10] 00:06:02, metric 2
                 > to 10.0.2.2 via em3.0
7.7.7.7/32      * [OSPF/10] 00:06:02, metric 2
                 > to 10.0.2.2 via em3.0
10.0.2.0/30     * [Direct/0] 00:06:02
                 > via em3.0
10.0.2.1/32     * [Local/0] 01:43:53
                 Local via em3.0
10.0.2.4/30     * [Direct/0] 01:44:16
                 > via em2.0
10.0.2.5/32     * [Local/0] 01:44:16
                 Local via em2.0
10.0.2.8/30     * [OSPF/10] 00:06:02, metric 2
                 > to 10.0.2.6 via em2.0
                 to 10.0.2.2 via em3.0
10.0.4.0/30     * [Direct/0] 01:45:06
                 > via em1.0
10.0.4.2/32     * [Local/0] 01:45:06
                 Local via em1.0
10.0.4.4/30     * [OSPF/10] 01:15:51, metric 2
                 to 10.0.4.13 via em0.0
                 > to 10.0.4.1 via em1.0
10.0.4.12/30    * [Direct/0] 01:45:32
                 > via em0.0
10.0.4.14/32    * [Local/0] 01:45:32
                 Local via em0.0
10.0.8.0/30     * [OSPF/10] 00:06:02, metric 3
                 > to 10.0.2.2 via em3.0
10.0.8.4/30     * [OSPF/10] 00:06:02, metric 2
                 > to 10.0.2.2 via em3.0
```

Slika 6.2.3.2. Prikaz tabele rutiranja za ruter R3

Iz tabele rutiranja možemo da vidimo preko kojih interfejsa je posmatrani ruter u komunikaciji sa drugim ruterima, i na koji način je dobio informaciju o ovoj vezi. Takođe, pored svake rute stoji odgovarajuća metrika. U tabeli rutiranja su samo najkraće rute od svih rutera u mreži doposmatranogrutera. Oznaka *Local* označava lokalni, tj. sopstveni interfejs, dok oznaka *Direct* predstavlja interfejs sa kojim ima direktnu vezu. Oznaka *OSPF* pokazuje rute koje je naučio posmatrani ruter pomoću ovog protokola. Do adresa do kojih postoji više putanja sa istom metrikom, navode se obe, a oznakom > pokazuje se je izabrana kao optimalnija. Kao što je ranije navedeno podešavanja ostalih rutera su data u priložima na kraju rada.

7. TESTIRANJE KONFIGURACIJE

Nakon konfigurisanja interfejsa i podizanja OSPF protokola rutiranja, komandom **show route**, kojom možemo videti tabelu rutiranja, možemo videti putanje koje će biti izabrane ka drugim ruterima u mreži. Na slici 7.1 je prikazana tabela rutiranja rutera R1.



```
gns3 - GNS3
t View C R1
1.1.1.1/32      *[Direct/0] 01:14:57
                > via lo0.0
2.2.2.2/32      *[OSPF/10] 00:55:46, metric 1
                > to 10.0.4.6 via em0.0
3.3.3.3/32      *[OSPF/10] 00:02:55, metric 1
                > to 10.0.4.14 via eml.0
4.4.4.4/32      *[OSPF/10] 00:01:58, metric 2
                > to 10.0.4.14 via eml.0
5.5.5.5/32      *[OSPF/10] 00:48:24, metric 2
                > to 10.0.4.14 via eml.0
6.6.6.6/32      *[OSPF/10] 00:00:15, metric 3
                > to 10.0.4.14 via eml.0
7.7.7.7/32      *[OSPF/10] 00:41:09, metric 3
                > to 10.0.4.14 via eml.0
10.0.2.0/30     *[OSPF/10] 00:48:47, metric 2
                > to 10.0.4.14 via eml.0
10.0.2.4/30     *[OSPF/10] 00:54:39, metric 2
                > to 10.0.4.14 via eml.0
10.0.2.8/30     *[OSPF/10] 00:53:05, metric 3
                > to 10.0.4.14 via eml.0
10.0.4.0/30     *[OSPF/10] 00:54:39, metric 2
                to 10.0.4.6 via em0.0
                > to 10.0.4.14 via eml.0
10.0.4.4/30     *[Direct/0] 01:30:15
                > via em0.0
10.0.4.5/32     *[Local/0] 01:30:15
                Local via em0.0
10.0.4.12/30    *[Direct/0] 01:29:17
                > via eml.0
10.0.4.13/32    *[Local/0] 01:29:17
                Local via eml.0
10.0.8.0/30     *[OSPF/10] 00:48:42, metric 4
                > to 10.0.4.14 via eml.0
10.0.8.4/30     *[OSPF/10] 00:48:42, metric 3
                > to 10.0.4.14 via eml.0
10.0.8.8/30     *[OSPF/10] 00:48:42, metric 3
                > to 10.0.4.14 via eml.0
224.0.0.5/32    *[OSPF/10] 00:58:07, metric 1
                MultiRecv
root>
```

Slika 7.1. Prikaz tabele rutiranja rutera R1

Sa slike 7.1 može se videti da, sve *loopback* interfejse u mreži, osim sopstvenog je ruter naučio posredstvom OSPF protokola.

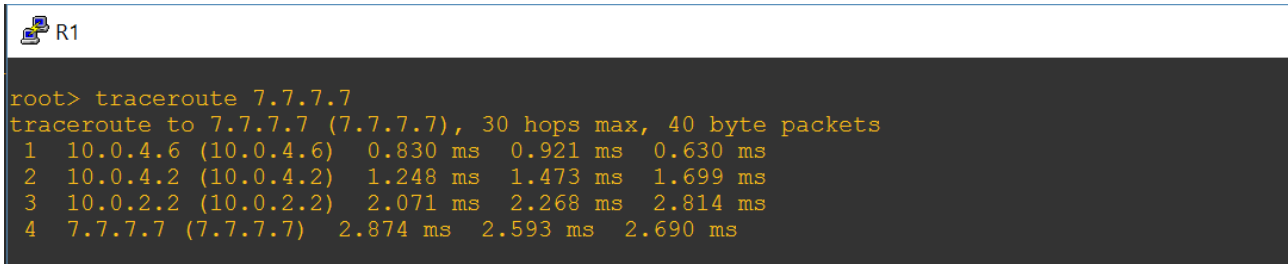
Nadalje, vide se ostali interfejsi mrežnih uređaja na kojima je aktivan OSPF porotokol, do kojih je ruta naučena, što direktno, što posredstvom OSPF protokola.

Kao što je navedeno ranije u radu, posmatraćemo tri moguća scenarija nakon kojih dolazi do promene putanja u mreži.

U primarnom stanju, tj. u stanju u kome su svi linkovi aktivni, optimalna putanja od oblasti tri (0.0.0.3) ka oblasti četiri (0.0.0.4) kroz *Backbone* oblast će biti preko mreže 10.0.2.0 tj. preko interfejsa *em3* rutera R3 i interfejsa *em0* rutera.

Od rutera R1 iz oblasti tri ka nekom od rutera iz oblasti četiri, optimalna putanja je preko rutera R2, a ne preko direktnog linka ka ABR ruteru R3, zbog podešene vrednosti metrike na tom linku na vrednost 5.

Pretpostavljenu putanju možemo videti zadavanjem komande *traceroute* u komandnom interfejsu rutera R1. Kao adresa do koje želimo da vidimo putanju uzeta je adresa *loopback* interfejsa rutera R7. Rezultat možemo videti na slici 7.2.



```
root> traceroute 7.7.7.7
traceroute to 7.7.7.7 (7.7.7.7), 30 hops max, 40 byte packets
 1  10.0.4.6 (10.0.4.6)  0.830 ms  0.921 ms  0.630 ms
 2  10.0.4.2 (10.0.4.2)  1.248 ms  1.473 ms  1.699 ms
 3  10.0.2.2 (10.0.2.2)  2.071 ms  2.268 ms  2.814 ms
 4  7.7.7.7 (7.7.7.7)  2.874 ms  2.593 ms  2.690 ms
```

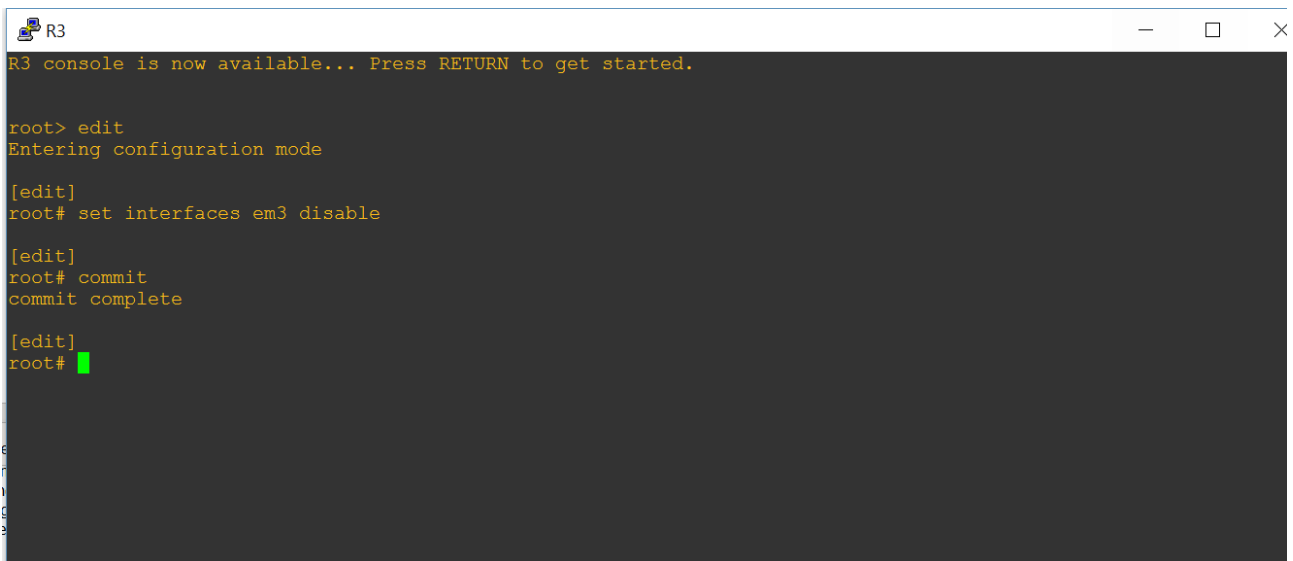
Slika 7.2. Prikaz rezultata naredbe *traceroute* rutera R1 u funkcionalnoj mreži

Osim ove komande, takođe je moguće koristiti i komandu *ping* koja pokazuje da je konektivnost moguća, ali ne i putanju samog paketa kao *traceroute* komanda.

7.1. Slučaj prekida jednog od linkova ili pada pojedinačnog interfejsa

Ovaj slučaj ćemo simulirati isključivanjem interfejsa *em3* rutera R3. Posmatrajući mrežu i njenu konfiguraciju, po isključenju datog interfejsa, saobraćaj bi trebalo da, od oblasti tri ka oblasti četiri, ide preko rutera R4.

Interfejs isključujemo ulaskom u menjanje konfiguracije željenog interfejsa naredbom **edit interfaces em3**, i zadavanjem komande **disable**. Na slici 7.1.1. prikazano je isključivanje interfejsa *em3* rutera R3. Naredbom **commit** potvrđujemo zadatu izmenu.



```
R3 console is now available... Press RETURN to get started.

root> edit
Entering configuration mode

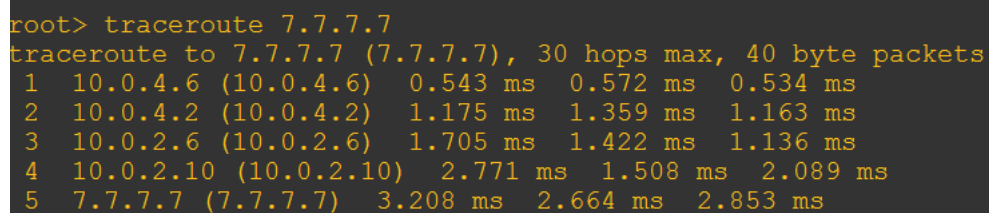
[edit]
root# set interfaces em3 disable

[edit]
root# commit
commit complete

[edit]
root#
```

Slika 7.1.1. Prikaz isključivanja interfejsa *em3* rutera R3

Nakon ovoga, zadavanjem komande **traceroute** na ruteru R1 i zadavanjem adrese *loopback* interfejsa rutera R7, kao što smo učinili u slučaju funkcionalne mreže vidimo novu putanju paketa.



```
root> traceroute 7.7.7.7
traceroute to 7.7.7.7 (7.7.7.7), 30 hops max, 40 byte packets
 1 10.0.4.6 (10.0.4.6) 0.543 ms 0.572 ms 0.534 ms
 2 10.0.4.2 (10.0.4.2) 1.175 ms 1.359 ms 1.163 ms
 3 10.0.2.6 (10.0.2.6) 1.705 ms 1.422 ms 1.136 ms
 4 10.0.2.10 (10.0.2.10) 2.771 ms 1.508 ms 2.089 ms
 5 7.7.7.7 (7.7.7.7) 3.208 ms 2.664 ms 2.853 ms
```

Slika 7.1.2. Prikaz rezultata komande **traceroute** rutera R1 ka ruteru R7 u slučaju pada linka između rutera R3 ka ruteru R5

Takođe, u ovom slučaju ostaće aktivne samo tri BFD sesije, kao što je prikazano na slici 7.1.3.

R3

```
commit complete

[edit]
root# set interfaces em3 disable

[edit]
root# commit
commit complete

[edit]
root# exit
Exiting configuration mode

root> show bfd session

Address          State      Interface    Detect    Transmit
10.0.2.6         Up        em2.0        Time     Interval  Multiplier
10.0.4.1         Up        em1.0        1.200    0.300    4
10.0.4.13        Up        em0.0        1.200    0.300    4

3 sessions, 3 clients
Cumulative transmit rate 10.0 pps, cumulative receive rate 10.0 pps
```

Slika 7.1.3. Prikaz aktivnih BFD sesija nakon onesposobljavanja em3 interfejsa rutera R3

Interfejs ćemo osposobiti zadavanjem komande **delete disable**, kao što je prikazano na slici 7.1.4. Naredbom **commit** potvrđujemo zadatu izmenu. Nakon ovoga OSPF proces se ponovo pokreće na ovom interfejsu i BFD sesija se ponovo uspostavlja.

R3

```
[edit]
root# delete interfaces em3 disable

[edit]
root# commit
commit complete

[edit]
root# exit
Exiting configuration mode

root> show bfd session

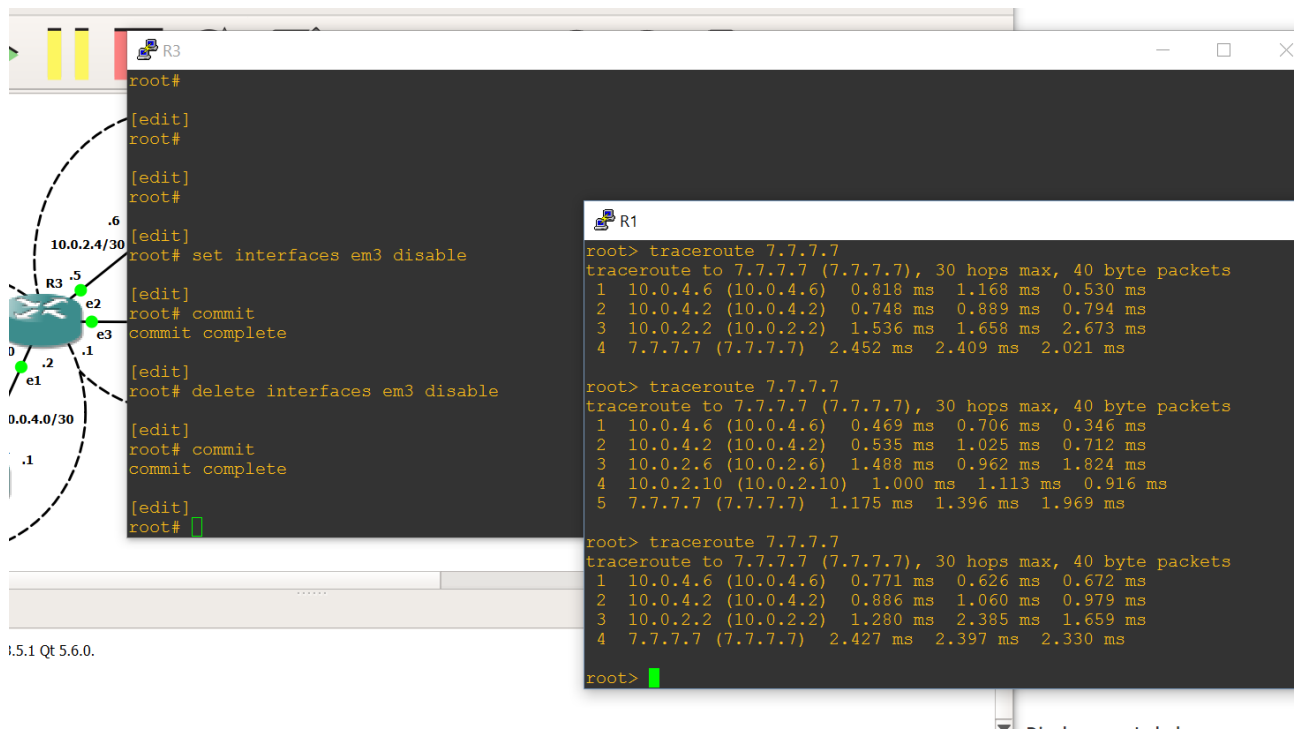
Address          State      Interface    Detect    Transmit
10.0.2.2         Down       em3.0        0.000    1.000    4
10.0.2.6         Up        em2.0        1.200    0.300    4
10.0.4.1         Up        em1.0        1.200    0.300    4
10.0.4.13        Up        em0.0        1.200    0.300    4

4 sessions, 4 clients
Cumulative transmit rate 11.0 pps, cumulative receive rate 10.0 pps

root> █
```

Slika 7.1.4. Uklanjanje komande *disable* na interfejsu em3 rutera R3 i ponovno uspostavljanje BFD sesije

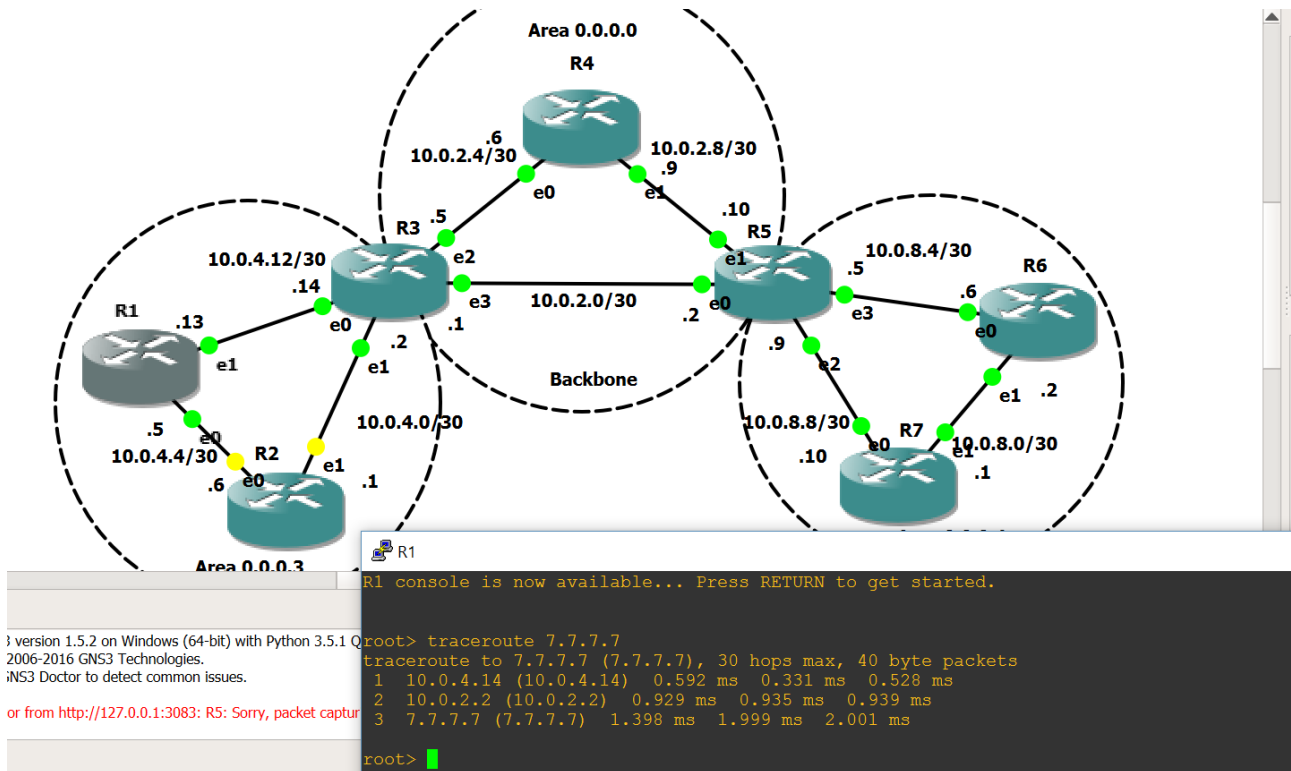
Nakon ovoga, mreža se ponovo vraća u prethodno stanje, i saobraćaj ponovo ide preko rutera R4. Konvergencija u prvobitno stanje prikazana je na slici 7.1.5.



Slika 7.1.5. Konvergencija u prvobitno stanje nakon aktiviranja interfejsa em3 rutera R3

7.2. Prestanak rada jednog od rutera R1, R2, R4, R6 i R7

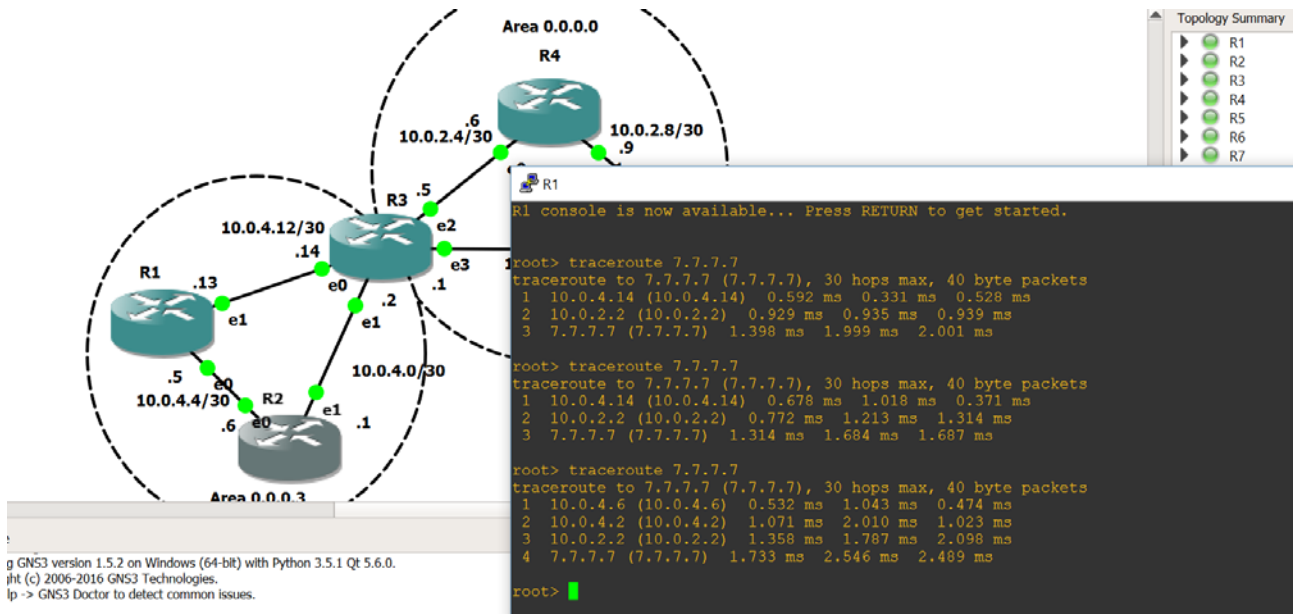
U ovom slučaju svi servisi takođe nastavljaju neometano da funkcionišu. Ovaj slučaj simuliraćemo biranjem opcije *suspend* za ruter R2 u okviru GNS3 simulatora, kao što je prikazano na slici 7.2.1.



Slika 7.2.1. Suspendovanje rutera R2 i naredba `traceroute` od R1 ka R7

Zbog konfigurisane metrike na interfejsu `em1` rutera R1 na vrednost 5, a ostavljene podrazumevane metrike na interfejsu `em0` koja iznosi 1, saobraćaj je ka oblasti tri (0.0.0.3) išao preko rutera R2. Nakon njegovog suspendovanja, saobraćaj ide preko putanje sa većom metrikom, jer je jedina postojeća.

Nakon prekida suspenzije rada rutera R2, mreža konvergira u prvobitno stanje. Ovo možemo videti ponovnim zadavanjem komande `traceroute`, što je prikazano na slici 7.2.2.

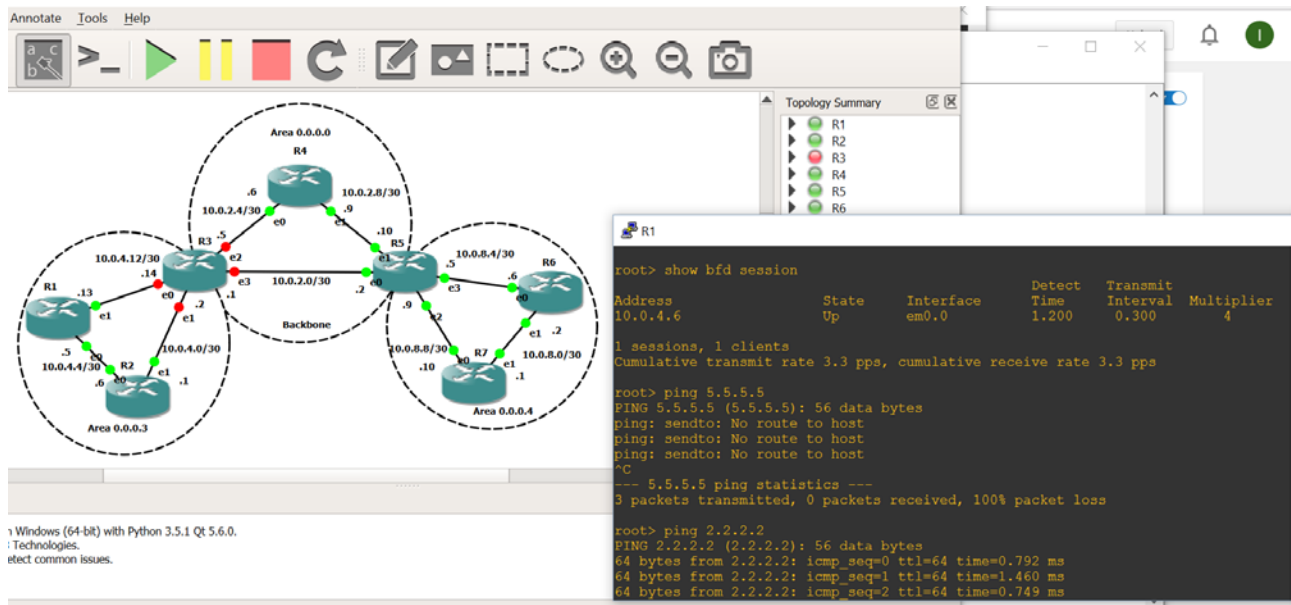


Slika 7.2.2. Konvergencija u prvobitno stanje nakon prekida suspenzije rada rutera R2

7.3. Prestanak rada jednog od ABR (*Area Border Router*) rutera

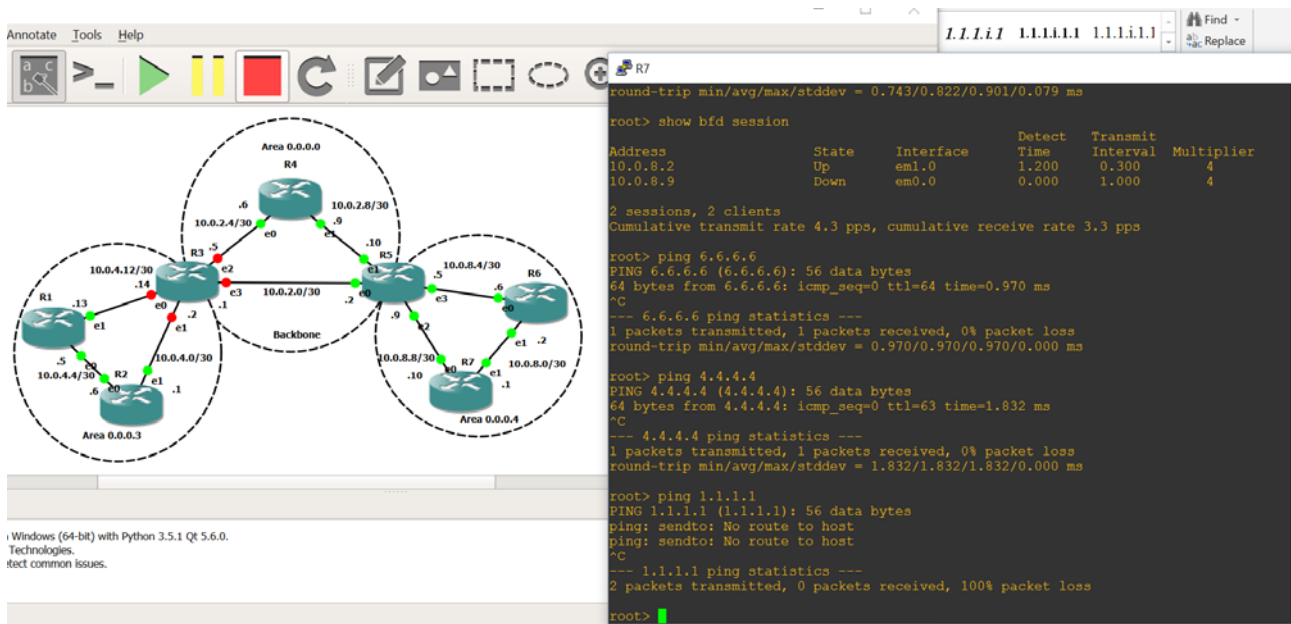
Prestankom rada rutera R3 ili R5, ruteri u pojedinačnim oblastima mreže će nastaviti da komuniciraju na nivou oblasti, ali će komunikacija između ovih oblasti biti u prekidu.

Ovaj scenario ćemo simulirati gašenjem rutera R3 u okviru GNS3 simulatora. Ovim oblast tri (0.0.0.3) biva „odsečena“ od *Backbone* oblasti, pa samim tim i od oblasti četiri (0.0.0.4). I dalje postoji komunikacija unutar oblasti tri (0.0.0.3), tj. između ova dva rutera. Takođe, BFD sesija na interfejsu *em1* više nije uspostavljena. Ovo možemo videti na slici 7.3.1.



Slika 7.3.1. Prikaz rezultata ping komande od strane rutera R1 u slučaju prestanka rada rutera R3

Između oblasti četiri i *backbone* oblasti ne dolazi do prekida komunikacije, kao što je prikazano na slici 7.3.2.



Slika 7.3.2. Prikaz rada naredbe ping u slučaju prestanka rada rutera R3

8. ZAKLJUČAK

Sve veća zastupljenost IP tehnologije u našoj svakodnevici povlači sa sobom njen razvoj i konstantnu adaptaciju na nove servise. Javlja se potreba za visokom dostupnošću mreže i njenim brzim oporavkom u slučaju kvara. Protokoli rutiranja, koji omogućavaju prenos informacija o stanju mreže u ovakvim tehnološkim ekosistemima, teže da budu što fleksibilniji. Fleksibilnost se ogleda u više aspekata. Tržišni aspekt podrazumeva postojanje javnih protokola rutiranja, tj. protokola koji funkcionišu na opremi različitih proizvođača. U ovom radu prikazan je rad OSPF protokola na ruterima kompanije Juniper. Infrastrukturni aspekt podrazumeva koncepte virtuelizacije, visoke raspoloživosti i konvergencije. U ovom radu, prikazani su upravo ovi koncepti. Tehnikom virtuelizacije kreirana je testna topologija koja se može susresti u praksi. Simulirana je raspoloživost mreže, a samim tim i servisa u njoj, u više različitih scenarija kvarova u mreži.

Poseban akcenat je stavljen na koncept mrežne konvergencije. OSPF protokol je dizajniran kao *link-state* protokol i samim tim pruža bržu konvergenciju u odnosu na *distance-vector* i *path-vector* protokole. Osim toga, ovaj protokol ima podršku za *fast-hello* mehanizam, koji, u odnosu na vremenski minimum od jedne sekunde za standardne *hello* pakete, omogućava detekciju prekida linka u intervalima kraćim od jedne sekunde.

Alternativu primeni standardnih *fast-hello* mehanizama pruža BFD protokol, protokol koji je se implementira na hijerarhijski nižoj ravni od standardnih protokola rutiranja i nezavisan je pogledu njihovog izbora. Ovaj protokol ima opciju funkcionisanja na dve ravni, kontrolnoj i ravni prosleđivanja. S obzirom da je rad rađen u emuliranoj sredini, sagledana je samo implementacija na kontrolnoj ravni.

Ovim radom prikazana je implementacija navedenih koncepata. Objasnjen je način kreiranja virtuelizovane sredine za testiranje, i prikazan rezultat testiranja različitih scenarija. Dodatno, kao prilog je data konfiguracija mrežnih uređaja, kao i adresna šema njihovih interfejsa što bi čitaocu trebalo da omogući jednostavno ponavljanje testova urađenih u ovoj tezi.

LITERATURA

- [1] D. Katz, D. Ward, Request for Comments: 5880, Bidirectional Forwarding Detection (BFD), <https://tools.ietf.org/pdf/rfc5880.pdf>
- [2] J. Moy, Request for Comments: 2328, OSPF v2, <https://www.ietf.org/rfc/rfc2328.txt>
- [3] <http://blog.ine.com/2010/06/02/ospf-fast-convergency/>
- [4] Rahul Aggarwal, Applications of Bidirectional Forwarding Detection (BFD), <http://meetings.ripe.net/ripe-48/presentations/ripe48-eof-bfd.pdf>
- [5] http://www.juniper.net/documentation/en_US/junos14.2/topics/topic-map/ospf-bfd.html
- [6] J. Neumann, The book of GNS3, No Strach Press, Inc., San Francisco 2015.
- [7] N. Krajinović, Rutiranje-osnove, Elektrotehnički fakultet, <http://telekomunikacije.etf.bg.ac.rs/predmeti/ot4ai/rutiranje-osnove.pdf>

A. PRILOGA

A.1. Konfiguracija rutera R1

```
## Last commit: 2016-09-21 00:40:18 UTC by root
version 12.1R1.9;
system {
  root-authentication {
    encrypted-password "$1$TP5PHUOR$gAPQLCeh5PRLJVhemDNrv."; ## SECRET-DATA
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any notice;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
}
interfaces {
  em0 {
    unit 0 {
      family inet {
        address 10.0.4.5/30;
      }
    }
  }
  em1 {
    unit 0 {
      family inet {
```

```
        address 10.0.4.13/30;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 1.1.1.1/32;
        }
    }
}
protocols {
    ospf {
        area 0.0.0.3 {
            interface em0.0 {
                metric 1;
                bfd-liveness-detection {
                    minimum-interval 300;
                    multiplier 4;
                    full-neighbors-only;
                }
            }
            interface em1.0 {
                metric 5;
                bfd-liveness-detection {
                    minimum-interval 300;
                    multiplier 4;
                    full-neighbors-only;
                }
            }
            interface lo0.0;
        }
    }
}
```

A.2. Konfiguracija rutera R2

```
## Last commit: 2016-09-20 23:29:20 UTC by root
version 12.1R1.9;
system {
  root-authentication {
    encrypted-password "$1$J07Uyifo$mbBLa4kuwTLhyc1/btQtN/"; ## SECRET-DATA
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any notice;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
}
interfaces {
  em0 {
    unit 0 {
      family inet {
        address 10.0.4.6/30;
      }
    }
  }
  em1 {
    unit 0 {
      family inet {
        address 10.0.4.1/30;
      }
    }
  }
}
```

```

lo0 {
  unit 0 {
    family inet {
      address 2.2.2.2/32;
    }
  }
}
protocols {
  ospf {
    area 0.0.0.3 {
      interface em0.0 {
        bfd-liveness-detection {
          minimum-interval 300;
          multiplier 4;
          full-neighbors-only;
        }
      }
      interface em1.0 {
        bfd-liveness-detection {
          minimum-interval 300;
          multiplier 4;
          full-neighbors-only;
        }
      }
      interface lo0.0;
    }
  }
}

```

A.3. Konfiguracija rutera R3

```

## Last commit: 2016-09-21 00:46:19 UTC by root
version 12.1R1.9;
system {
  root-authentication {
    encrypted-password "$1$/Hisp4EL$9AP.dP.ajXw/Y7V3UcbNy/"; ## SECRET-DATA
  }
}

```

```
syslog {
  user * {
    any emergency;
  }
  file messages {
    any notice;
    authorization info;
  }
  file interactive-commands {
    interactive-commands any;
  }
}
}
interfaces {
  em0 {
    unit 0 {
      family inet {
        address 10.0.4.14/30;
      }
    }
  }
  em1 {
    unit 0 {
      family inet {
        address 10.0.4.2/30;
      }
    }
  }
  em2 {
    unit 0 {
      family inet {
        address 10.0.2.5/30;
      }
    }
  }
  em3 {
    unit 0 {
      family inet {
```



```

        address 10.0.2.1/30;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 3.3.3.3/32;
        }
    }
}
protocols {
    ospf {
        area 0.0.0.3 {
            interface em0.0 {
                bfd-liveness-detection {
                    minimum-interval 300;
                    multiplier 4;
                    full-neighbors-only;
                }
            }
            interface em1.0 {
                bfd-liveness-detection {
                    minimum-interval 300;
                    multiplier 4;
                    full-neighbors-only;
                }
            }
        }
        area 0.0.0.0 {
            interface em2.0 {
                bfd-liveness-detection {
                    minimum-interval 300;
                    multiplier 4;
                    full-neighbors-only;
                }
            }
        }
    }
}

```

```
interface em3.0 {  
    bfd-liveness-detection {  
        minimum-interval 300;  
        multiplier 4;  
        full-neighbors-only;  
    }  
}  
interface lo0.0;  
}  
}
```

A.4. Konfiguracija rutera R4

```
## Last commit: 2016-09-21 00:03:55 UTC by root
version 12.1R1.9;
system {
  root-authentication {
    encrypted-password "$1$3LTHdm.9$gc3HmdosJbnTg15seNYm1"; ## SECRET-DATA
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any notice;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
}
interfaces {
  em0 {
    unit 0 {
      family inet {
        address 10.0.2.6/30;
      }
    }
  }
  em1 {
    unit 0 {
      family inet {
        address 10.0.2.9/30;
      }
    }
  }
}
```

```

lo0 {
  unit 0 {
    family inet {
      address 4.4.4.4/32;
    }
  }
}
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface em0.0 {
        bfd-liveness-detection {
          minimum-interval 300;
          multiplier 4;
          full-neighbors-only;
        }
      }
      interface em1.0 {
        bfd-liveness-detection {
          minimum-interval 300;
          multiplier 4;
          full-neighbors-only;
        }
      }
      interface lo0.0;
    }
  }
}
}

```

A.5. Konfiguracija rutera R5

```

## Last commit: 2016-09-20 23:15:53 UTC by root
version 12.1R1.9;
system {
  root-authentication {
    encrypted-password "$1$XgBGbTDq$6KAPLq8KWleAy8l5Bf6/21"; ## SECRET-DATA
  }
}

```

```
syslog {
  user * {
    any emergency;
  }
  file messages {
    any notice;
    authorization info;
  }
  file interactive-commands {
    interactive-commands any;
  }
}
}
interfaces {
  em0 {
    unit 0 {
      family inet {
        address 10.0.2.2/30;
      }
    }
  }
  em1 {
    unit 0 {
      family inet {
        address 10.0.2.10/30;
      }
    }
  }
  em2 {
    unit 0 {
      family inet {
        address 10.0.8.9/30;
      }
    }
  }
  em3 {
    unit 0 {
      family inet {
```

```
        address 10.0.8.5/30;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 5.5.5.5/32;
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface em0.0;
            interface em1.0;
        }
        area 0.0.0.4 {
            interface em2.0;
            interface em3.0;
            interface lo0.0;
        }
    }
}
}
```

A.6. Konfiguracija rutera R6

```
## Last commit: 2016-09-21 00:05:40 UTC by root
version 12.1R1.9;
system {
    root-authentication {
        encrypted-password "$1$9MU8QUHh$JRGFE.N1wejehn758p/0M0"; ## SECRET-DATA
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
interfaces {
    em0 {
        unit 0 {
            family inet {
                address 10.0.8.6/30;
            }
        }
    }
    em1 {
        unit 0 {
            family inet {
                address 10.0.8.2/30;
            }
        }
    }
}
```

```

lo0 {
  unit 0 {
    family inet {
      address 6.6.6.6/32;
    }
  }
}
protocols {
  ospf {
    area 0.0.0.4 {
      interface em0.0 {
        bfd-liveness-detection {
          minimum-interval 300;
          multiplier 4;
          full-neighbors-only;
        }
      }
      interface em1.0 {
        bfd-liveness-detection {
          minimum-interval 300;
          multiplier 4;
          full-neighbors-only;
        }
      }
      interface lo0.0;
    }
  }
}

```

A.7. Konfiguracija rutera R7

```

version 12.1R1.9;
system {
  root-authentication {
    encrypted-password "$1$U40y7Kcy$IeH5izPFWJYFYt4iH13v20"; ## SECRET-DATA
  }
  syslog {

```



```
user * {
    any emergency;
}
file messages {
    any notice;
    authorization info;
}
file interactive-commands {
    interactive-commands any;
}
}
interfaces {
    em0 {
        unit 0 {
            family inet {
                address 10.0.8.10/30;
            }
        }
    }
    em1 {
        unit 0 {
            family inet {
                address 10.0.8.1/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 7.7.7.7/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.4 {
```

```

interface em0.0 {
    bfd-liveness-detection {
        minimum-interval 300;
        multiplier 4;
        full-neighbors-only;
    }
}
interface em1.0 {
    bfd-liveness-detection {
        minimum-interval 300;
        multiplier 4;
        full-neighbors-only;
    }
}
interface lo0.0;
}
}
}
interface em1.0 {
    bfd-liveness-detection {
        minimum-interval 300;
        multiplier 4;
        full-neighbors-only;
    }
}
}
}
area 0.0.0.0 {
    interface em2.0 {
        bfd-liveness-detection {
            minimum-interval 300;
            multiplier 4;
            full-neighbors-only;
        }
    }
}
interface em3.0 {
    bfd-liveness-detection {
        minimum-interval 300;
        multiplier 4;
        full-neighbors-only;
    }
}

```

```
    }  
  }  
  interface lo0.0;  
  }  
  }  
}
```

B. ADRESNA ŠEMA TESTNE TOPOLOGIJE

Ruter	interfejs	ip adresa	subnet mask	OSPF area
R1	em0	10.0.4.5	255.255.255.252	0.0.0.3
	em1	10.0.4.13	255.255.255.252	0.0.0.3
	lo0	1.1.1.1	255.255.255.255	0.0.0.3
R2	em0	10.0.4.6	255.255.255.252	0.0.0.3
	em1	10.0.4.1	255.255.255.252	0.0.0.3
	lo0	2.2.2.2	255.255.255.255	0.0.0.3
R3	em0	10.0.4.14	255.255.255.252	0.0.0.3
	em1	10.0.4.2	255.255.255.252	0.0.0.3
	em2	10.0.2.5	255.255.255.252	0.0.0.0
	em3	10.0.2.1	255.255.255.252	0.0.0.0
	lo0	3.3.3.3	255.255.255.255	0.0.0.0
R4	em0	10.0.2.6	255.255.255.252	0.0.0.0
	em1	10.0.2.9	255.255.255.252	0.0.0.0
	lo0	4.4.4.4	255.255.255.255	0.0.0.0
R5	em0	10.0.2.2	255.255.255.252	0.0.0.0
	em1	10.0.2.10	255.255.255.252	0.0.0.0
	em2	10.0.8.9	255.255.255.252	0.0.0.4
	em3	10.0.8.5	255.255.255.252	0.0.0.4
	lo0	5.5.5.5	255.255.255.255	0.0.0.4
R6	em0	10.0.8.6	255.255.255.252	0.0.0.4
	em1	10.0.8.2	255.255.255.252	0.0.0.4
	lo0	6.6.6.6	255.255.255.255	0.0.0.4
R7	em0	10.0.8.10	255.255.255.252	0.0.0.4
	em1	10.0.8.1	255.255.255.252	0.0.0.4
	lo0	7.7.7.7	255.255.255.255	0.0.0.4

Tabela B.1. Adresna šema interfejsa rutera u testnoj topologiji