UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE I MEĐUNARODNU SARADNJU

UNAPREĐENJE PERFORMANSI OBRAZOVNIH SISTEMA PRIMENOM BEŽIČNIH MESH MREŽA

- DOKTORSKA DISERTACIJA-

Mentor:
prof. dr Mladen Veinović

Kandidat:
Aleksandar Zakić, master

BEOGRAD, 2015.

UNIVERSITY OF SINGIDUNUM
BELGRADE
DEPARTMANT OF POSTGRADUATE STUDIES AND INTERNATIONAL COOPERATION DOCTORAL

IMPROVING PERFORMANCE LEARNING SYSTEM USING WIRELESS MESH NETWORK

- DOCTORAL DISSERTATION-

Mentor: **Candidate:**

prof. Mladen Veinović, PhD **Aleksandar Zakić, master**

BELGRADE, 2015.

SAŽETAK

Nastava iz oblasti bežičnog umrežavanja i mobilnih mreža predstavlja neophodnu oblast bez koje današnje savremeno prihvatanja znanja iz računarskih nauka ne bi moglo da se sprovede. Studenti uče apstraktne pojmove o radu ovih sistema, ali se i svakodnevno susreću s različitim tehnologijama umrežavanja. Jedan od novih načina za povezivanje apstraktnih koncepata jeste korišćenje savremenih tehnologija kao što su računarstvo u oblaku (Cloud computing), emulatori i virtualizacija. Suština unapređenja performansi obrazovnih sistema se ogleda u izgradnji distribuiranih aplikacija primenom savremenih tehnologija. Za izgradnju ovakvog okruženja potrebno je da se studenti i predavači aktivno uključe u proces razvoja novih tehnologija, da bi stvorili optimalne uslove za eksperimente i nova naučna ostvarenja u oblasti informacionih i komunikacionih tehnologija.

ABSTRACT

Teaching in the field wireless and mobile network represent necessary area without which today society and acceptance of computer knowledge can't be conducted. Students learn abstract notions about this systems, but they also on daily basis come across different technologies of networking. One of the newest ways for connecting abstract concepts is use of modern technologies like Cloud computing, emulators and virtualization. The essence of improvement performance of education systems is shown through building distributed applications by using modern technology. For this type of surrounding's, students and teachers need to be active in the process of development of new technologies, so they can make optimal conditions for experiments and new scientific achievements in information-communication technologies.

INDEKS SLIKA

INDEKS TABELA

REČNIK POJMOVA

U daljem tekstu date su definicije osnovnih pojmova:

MESH – tehnologija umrežavanja i fizička topologija. Pogodnost ove tehnologije jeste ta što se veoma lako nadograđuje postojeća mrežna infrastruktura.

WMN (Wireless Mesh Network) – bežična mesh mreža. Otpornost na kvarove i jednostavno proširenje su glavne prednosti WMN mreža. Sama topologija, odnosno veliki broj čvorova čini WMN mreže otpornim na kvarove.

AP (Access Point) – bežična pristupna tačka. Uređaj koji spaja WiFi uređaje u jednu bežičnu mrežu. AP se obično spaja na žičanu mrežu i prenosi podatke između žičanih i bežičnih mreža.

WiFi – Wireless-Fidelity (IEEE 802.11) jeste bežična mreža gde se podaci između dva ili više računara prenose radio-frekvencijama (RF) i odgovarajućim antenama. Najčešće se koristi u LAN mrežama (WLAN), ali je u poslednje vreme sve zastupljeniji bežični pristup WAN mreži – internetu. Wi-Fi je brend Wi-Fi alijanse koja propisuje standarde i izdaje sertifikate za sve Wi-Fi uređaje. Wi-Fi je 1991 godine izumela NCR korporacija/AT&T u NieuwegeinuNivegejnu, Holandija.

WLAN – WLAN mreža predstavlja nov način povezivanja računara bez kabla. Primenjuje se na mestima gde kablovi predstavljaju fizičku smetnju ili tamo gde se ukaže potreba za ovakvom vrstom mreže. Da bi ovakva mreža funkcionisala potrebno je da postoji optička vidljivost između računara i razvodnika, u zavisnosti koji se tip mreže koristi.

IEEE 802.11 – Skup standarda za bežične lokalne mreže (WLAN) računara za kućne i kancelarijske potrebe. Najpopularnije su definisane standardom IEEE 802.11b i 802.11g.

Routing – Protokol rutiranja (engl. Ruting protocol) predstavlja set pravila kojim ruteri dinamički razmenjuju informacije o rutama kojima se paket kreće kako bi stigao na željenu destinaciju. Kad se dogodi neka izmena u topologiji računarske mreže, najbliži ruter kod koga se desila promena je zapisuje u svoju tabelu rutiranja, a potom protokoli rutiranja pokreću mehanizme kojima se informacija o promeni u topologiji prosleđuje ostalim uređajima u mreži. Na ovaj način ruteri dinamički ažuriraju svoje tabele rutiranja. Ovi protokoli pripadaju sloju mreže referentnog OSI modela.

Ad hoc – Ad hok mreže formiraju korisnici koji žele da komuniciraju bez potrebe za infrastrukturom. Svaki modul u mreži ima mogućnost bežičnog komuniciranja i može u svakom trenutku pristupiti i napustiti mrežu. Zbog ograničenog broja modula koji

mogu istovremeno da ostvare komunikaciju u mreži koristi se multi-hop način veze. Ad hoc veza se upotrebljava u senzorskim mrežama, WMN mrežama, u vojnoj komunikaciji i kao proširenje i dopuna već postojećih mreža radi efikasnijeg delovanja.

IGW (Internet Gateway) – Internet gejtveji povezani s Internetom. Oni čine okosnicu infrastrukture za pružanje internet konekcije za elemente na drugom nivou.

MR (Mesh Router) – Entiteti na drugom nivou se nazivaju bežični *mesh* ruteri koji eliminišu potrebu za žičanom infrastrukturom; svaki MR prenosi saobraćaj u *multi-hop* režimu prema IGW.

MC (Mesh Clients) – mesh klijenti su bežični uređaji korisnik**a.**

MANET (Mobile Ad Hoc Networks) – samoorganizujući skup mreže mobilnih uređaja povezanih bežično.

SADRŽAJ

UVOD

Znanje predstavlja osnovu individualnog i društvenog razvoja. Učenje, kognitivni razvoj, istraživanje i izum stari su koliko i naša vrsta, ali izgradnja znanja nije. Proces stvaranja znanja odnosi se na promenu postojećeg znanja i kreiranje novog oblika, tzv. javnog znanja (communiry knowledge), koje se nalazi svuda oko nas, znanja koje mogu koristiti oni koji ga kreiraju, ali i drugi ljudi (Scardamalia, M., & Bereiter, C. 2010). Znanje danas nije samo novi koncept, već i stvaranje novog procesa učenja i istraživanja. Učenje se danas posmatra kao aktivno građenje znanja, kroz formu teoretske nastave i laboratorijskih vežbi. Takođe, jedno od glavnih obeležja stvaranja znanja jeste to da se ono dešava u kontekstu grupe, odnosno zahteva interakciju između članova grupe koja uči.

U preporukama IEEE za umrežene laboratorije definišu se metode za povezivanje objekata za učenje s dizajnom i implementacijom inteligentnih okruženja za učenje, zajedno sa metodama učenja i alatima neophodnim za sprovođenje aktivnosti praktičnog laboratorijskog rada. Neophodno je dizajnirati i implementirati okruženje za učenje na takav način gde je pedagoški kontekst uzet u obzir.

Da bi se dostigli ovi ciljevi razvijeno je softversko okruženje MeshLAB. MeshLAB se zasniva na trenutno aktuelnim tehnologijama računarstva u oblaku i virtualizacije. Osnovne funkcionalnosti sistema, kao i iskustva u primeni okruženja na bežične mesh mreže (WMN) opisani su u nastavku. Prikazane su laboratorijske vežbe bazirane na MeshLAB okruženju, a dobijeni podaci predstavljeni su matematičkim modelom iz spektralne teorije grafova na čijim teoremama se zasnivaju mesh mreže.

1.1. Predmet istraživanja

Nastava iz oblasti bežičnog umrežavanja i mobilnih mreža predstavlja neophodnu oblast bez koje današnje savremeno prihvatanja znanja iz računarskih nauka ne bi moglo da se sprovede. Studenti uče apstraktne pojmove o radu ovih sistema, ali se i svakodnevno susreću s različitim tehnologijama umrežavanja. Jedan od novih načina za povezivanje apstraktnih koncepata jeste i korišćenje savremenih tehnologija kao što su računarstvo u oblaku (Cloud computing), emulatori i virtualizacija. Ove tehnologije omogućavaju poboljšanje procesa učenja i proširivanje znanja studenata na lako razumljiv način jer pružaju mogućnost za izgradnju i testiranje različitih tipova tehnologija, protokola i servisa u bežičnim i mobilnim mrežama.

U poslednjih nekoliko godina simulatore zamenjuju emulatori koji pomoću virtualizacije imaju sposobnost oponašanja realnih mrežnih uređaja s različitim

tipovima protokola. Mrežni simulatori (ns-2, ns-3, OMNet i sl.) često daju različite rezultate prilikom testiranja protokola u odnosu na performanse realnih uređaja, pa su zbog toga potrebna nova istraživanja u domenu bežičnih mesh mreža (WMN) primenom virtualizacije, računarstva u oblaku i emulatora. Ovaj pristup omogućava stvaranje okruženja čiji rezultati u istraživanjima mogu da imaju doprinos pri testiranju WMN protokola rutiranja u različitim uslovima.

WMN mreže su tip mobilnih ad hok mreža (Mobile Ad hoc Network – MANET). Mobilne Ad hok mreže se sastoje od mobilnih čvorova koji mogu da se kreću proizvoljno, povezanih bežičnim linkovima, često bez postojeće infrastrukture ili fiksnih baznih stanica. Ovakve komunikacione mreže često se modeluju u obliku grafa.

Napredovanje sistema WMN mreža, posebno u oblastima novih tehnologija rutiranja, štednje energije i sve većih razmera ovih mreža, zahteva neprestanu nadgradnju okruženja, koja s druge strane traže sve više računarskih resursa, pa se zbog toga projektovanje ovako složenih emulatora javlja kao jedan od problema koji treba rešiti.

Wireless Mesh Network (WMN) je tehnologija u razvoju. Karakteriše je distribuirana struktura i samoorganizovanje. Osnovna topološka karakteristika WMN mreža jeste da postoji samo jedan ili nekoliko čvorova za priključenje na infrastrukturnu mrežu ili na gejtvej, a svi ostali čvorovi se povezuju preko suseda na izlaz (Internet).

U zavisnosti od tipa korisnika, mesh mreže mogu biti različite veličine i namene. Zato se mesh mreže mogu konstruisati na različite načine. Manje mesh mreže ne moraju da se razvijaju po standardu (npr. Gnutella, BitTorrent). Međutim, ako se mesh mreža razvija za veliki broj korisnika, tada se ona mora zasnivati na određenom standardu na osnovu koga će se usmeravati paketi kroz mrežu. Da bi više korisnika u isto vreme moglo da vrši prenos podataka, dodeljuju se kanali za prenos kroz čvorove. Većina današnjih mreža razvija se po određenom standardu. U ovom radu se opisuje IEEE 802.11 standard za mesh mreže, kao i protokoli rutiranja pomoću kojih se vrši razmena podataka kroz mesh mrežu.

Mesh mrežna topologija omogućava da svaki čvor prenosi podatke i služi takođe kao relej za druga čvorišta koja sarađuju prilikom propagacije podataka u mreži. Odnosno, mesh mreže se odlikuju nepostojanjem središnjeg čvora preko koga se odvija komunikacija, što znači da svaki čvor u mreži služi kao prenosilac podataka za druge čvorove. U odnosu na druge mrežne topologije (Star, Ring, Bus i dr.), mesh mreže su pouzdanije jer postoji više puteva (ruta) kojima se podaci mogu prenositi između korisnika. Druga prednost mesh mreža zbog koje one postaju sve popularnije jeste ta što mogu vrlo jednostavno da se sagrade u promenljivoj okolini jer se nadograđuju na postojeće komunikacijske mreže. Nadogradnja se svodi davanjem novih čvorova na postojeće komunikacijske mreže. Na osnovu toga mesh mreže se sve češće primenjuju

kao komunikacijske mreže u auto-industriji, avio-industriji i u fabričkim postrojenjima i akademskim ustanovama.

Princip rada i prenos podataka u mesh mrežama odvija se na osnovu različitih komunikacijskih standarda (IEEE 802.11, IEEE 802.16). Podaci u mesh mreži šalju se optimalnim putem, gde je osigurana zaštita od grešaka u prenosu podataka i gde je najveća brzina prenosa. Takav proces toka podataka kroz mesh mrežu naziva se usmeravanje ili rutiranje (routing). Postoji mnogo algoritama za usmeravanje paketa kroz mesh mreže, a najpoznatiji su AODV, B.A.T.M.A.N, DNVR, OLSR i dr.

Bežične mesh mreže (Wireless Mesh Network) se posebno brzo razvijaju. Na osnovu konstrukcije dele se na: ravne, hijerarhijske i hibridne WMN. Bežične mesh mreže pokazuju bolje performanse od standardnih mobilnih i WLAN mreža i zbog toga su sve češće u potrebi.

Mesh mreže su prilično sigurne, ali, uprkos tome, javljaju se razne sigurnosne slabosti. U radu se opisuju i mehanizam koji se koriste za zaštitu podataka u mesh mrežama, kao i najčešće sigurnosne ranjivosti i pretnje mesh mrežama. Mesh mreže će se u budućnosti sve više širiti i razvijati jer pružaju najbolje performanse u mreži, bezbednije su i lako se nadograđuju.

Emulator predstavlja virtuelnu mašinu koja simulira kompletan hardver, što omogućava gost OS da se izvršava na potpuno nezavisnom procesoru. Na ovaj način virtualni hardver oponaša pravi uređaj. Ovaj model omogućava: smanjenje troškova pri testiranju novih uređaja, aplikacija ili protokola u ovom slučaju; analizu dobijenih rezultata; prilagođavanje okruženja zahtevima koji su potrebni u kreiranju nekog modela; sposobnost simulacije različitih arhitektura i tipova hardvera; sposobnost simulacije različitih platformi (Android, iOS, OpenWRT, Windows Mobile).

Prelaskom na virtualizaciju stvoren je novi pristup na mrežnom sloju koji omogućava međusobno povezivanje virtuelnih mašina (VM). VM mreže su nametnule zahteve u umrežavanju koji tradicionalno nisu raspoloživi.

1.2. Polazne hipoteze

U izradi doktorske disertacije polazi se od generalne hipoteze: Novi način za unapređenje performansi obrazovnih sistema je razvoj distribuiranih aplikacija na osnovu emulatora i virtualizacije posebno za tu namenu;

Izvedene hipoteze:

- u razjašnjavanju apstraktnih pojmova na predavanjima, nastavnici se susreću s problemom testiranja različitih WMN protokola rutiranja i velikim brojem čvorova;

- mrežni simulatori (ns-2, ns-3, OMNet i sl.) često daju različite rezultate prilikom testiranja protokola u odnosu na performanse realnih uređaja;
- softverski sistemi zasnovani na računarstvu u oblaku omogućavaju projektovanja emulatora za testiranje protokola rutiranja;
- definisanjem matematičkog modela, studentima se omogućava lakši pristup apstraktnim pojmovima iz spektralne teorije grafovana čijim teoremama se zasnivaju mesh mreže;

1.3. Naučne metode istraživanja

Radom na disertaciji obuhvaćene su sledeće naučne metode istraživanja:

- sistematsko proučavanje domaće i inostrane literature iz oblasti disertacije;
- kritička analiza problema mrežnih simulatora;
- evaluacija primene virtualizacije za kreiranje edukativnog modela za testiranje protokola rutiranja u WMN mrežama;
- odrađivanje i analiziranje prednosti i mana simulacionih sistema i rezultata evaluacije;
- definisanje matematičkog modela u cilju daljeg izučavanja WMN protokola rutiranja;
- definicija analitičkog modela ponašanja emulatora za testiranje protokola rutiranja;
- verifikacija polaznih hipoteza o mogućnostima realizovanog softverskog okruženja na osnovu rezultata modelovanja i rezultata dobijenih na osnovu realizacije odgovarajućih emulatora u relevantnim oblastima WMN mreža.

1.4. Ciljevi istraživanja

Opšti cilj istraživanja ove disertacije je modelovanje edukativnog softverskog sistema za emulaciju bežičnih mesh mreža. Praktični cilj ovog istraživanja je dizajniranje i implementacija edukativnog softverskog sistema za testiranje protokola rutiranja u WMN mrežama. Na osnovu dosadašnjih naučnoistraživačkih iskustava iz ove oblasti, kao i istraživanja u ovoj doktorskoj disertaciji, predloženo je softversko rešenje za podršku nastavi iz bežičnih mreža. Osnovu okruženja za emulaciju WMN mreža čine komponente koje su razvijene u računarstvu u oblaku primenom virtualizacije.

1.5. Naučni doprinos

Naučni doprinos doktorske disertacije jeste u domenu analize i sinteze softverskog sistema koji treba da omogući kreiranje WMN mreža pogodnih za rad u edukativnom domenu. Kao sastavni delovi doktorske disertacije dati su sledeći naučni doprinosi:

- sistematizacija i klasifikacija postojećih rešenja u projektovanju emulatora iz oblasti WMN mreža;
- kreiranje metodologije projektovanja emulatora WMN mreža na osnovu klasifikovanih rešenja koja su bila primenjena, koja se primenjuju ili koja se mogu primeniti u okviru ove discipline;
- osnovni doprinos je predlog i implementacija novog edukativnog okruženja koji na jednostavan način treba da omogući kreiranje WMN mreža sposobnih za rad u edukativnom okruženju, za testiranje protokola rutiranja. Sistem obezbeđuje vizuelni pregled i izgradnju kompleksnog sistema WMN mreža od najnižeg nivoa, kao i mogućnostima povezivanja proizvoljnih modula pomoću alata;
- postavljeni analitički model opisuje ponašanje emulatora WMN mreža razvijenih prema predloženoj metodologiji prilikom rada u edukativnom okruženju, kao i matematički model primenom spektralne teorije grafova;

1.6. Plan istraživanja

Plan istraživanja obuhvatio je ispitivanje trenutnog stanja u oblasti projektovanja emulatora za WMN mreže, primenu virtualizacije i računarstva u oblaku, analizu nastave iz bežičnih mreža, sistematizaciju postojećih rešenja shodno uočenim potrebama, kritičku analizu prednosti i nedostataka raspoloživih softverskih okruženja, predlog rešenja uočenih problema, pregled karakteristika realizovanog softverskog okruženja i razvijanje analitičkog modela ponašanja primenom spektralne teorije grafova.

U doktorskoj disertaciji prikazan je pregled oblasti u nastavi iz bežičnih mreža na studijama računarskih nauka, kao i pregled oblasti računarstva u oblaku i virtualizacije. U nastavku je dat detaljan pregled i evaluacija raspoloživih emulatora sa stanovišta oblasti primene i karakteristika, kao i njihova analiza sa stanovišta moguće primene u edukativnom domenu. Prikazan je dizajn edukativnog modela koji treba da omogući razvoj WMN mreža proizvoljnog nivoa složenosti. Nakon toga je dat analitički model koji opisuje karakteristike realizovanog sistema prikazan matematičkim modelom iz spektralne teorije grafova. U sledećem delu prikazan je prototip emulatora razvijen prema opisanom postupku s ciljem da se omogući njegova primena u nastavi i u budućim istraživanjima WMN protokola rutiranja. Definisane su i laboratorijske vežbe koje su implementirane u softverskom okruženju.

1.7. Struktura rada

Doktorska disertacija sadrži osam poglavlja, skup neophodnih priloga i pregled korišćene literature, indeks slika i grafikona.

Prvo poglavlje predstavlja uvod u disertaciju. Opisuje značaj znanja kao osnovu za razvoj društva, metodologiju disertacije i analizu pojmova.

Drugo poglavlje predstavlja uvod u bežične mesh mreže i njihove karakteristike, evaluaciju WMN mreža i podelu na osnovu fizičke topologije.

Treće poglavlje bavi se WMN protokolima rutiranja, vrstama WMN protokola i standardima. Suština funkcionalnosti mesh mreža jeste u usmeravanju paketa *(Routing)*, pomoću koga se omogućava primanje i slanje podataka bilo gde u mreži.

Četvrto poglavlje opisuje trenutna istraživanja bezbednosnih protokola za WMN mreže koji se fokusira na dizajn mehanizama za različite tipove slabosti na različitim slojevima.

Peto poglavlje bavi se analizom performansi WMN protokola i implementacijom OLSR i B.A.T.M.A.N. protokola na realnim uređajima.

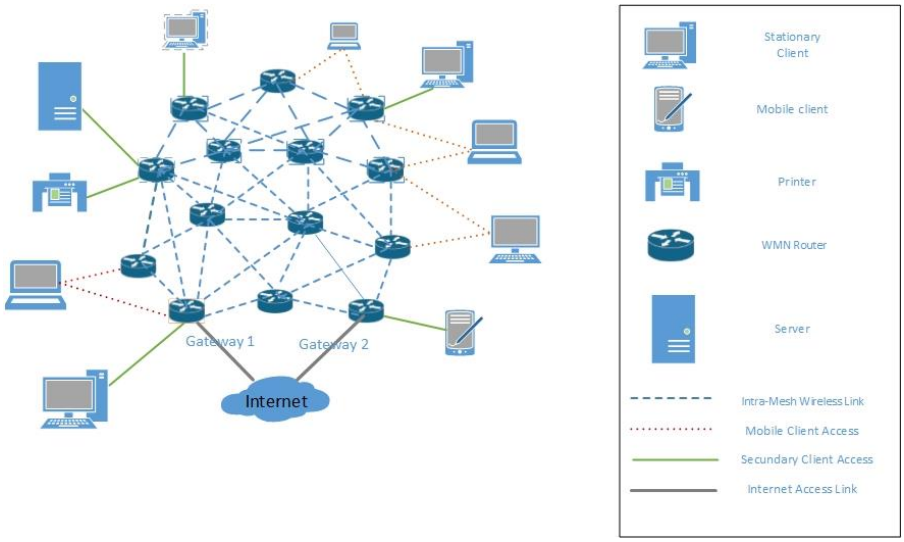
Šesto poglavlje sadrži opis emulatora i okruženja u kome se razvijao model za testiranje WMN protokola rutiranja i opis stvaranja radnog okruženja za testiranje različitih protokola za WMN mreže. U ovom poglavlju dat je detaljan pregled i evaluacija raspoloživih rešenja za kreiranje edukativnog modela. Opisuje se razlika između mrežnih simulatora i novog koncepta kreiranja emulatora pomoću računarstva u oblaku i virtualizacije.

Sedmo poglavlje sadrži matematički model iz spektralne teorije grafova i njegovu primenu u WMN mrežama. Ovo poglavlje opisuje primenu matrice susedstva i Laplasovu teoremu kreiranja matematičkog modela na osnovu parametara koji su dobijeni iz vežbi kreiranih pomoću emulatora MeshLAB (edukativnog modela za kreiranje WMN mreže). Proučavanjem primene spektralne teorije grafova u WMN mrežama javila se ideja o klasterizaciji čvorova (koja je obrazložena u disertaciji) i koja bi u budućim istraživanjima mogla da ima veliki doprinos u razvoju protokola rutiranja u WMN mrežama.

U osmom poglavlju je izložen zaključak. Na kraju su dati neophodni prilozi i pregled korišćene literature.

2. KARAKTERISTIKE WMN MREŽA

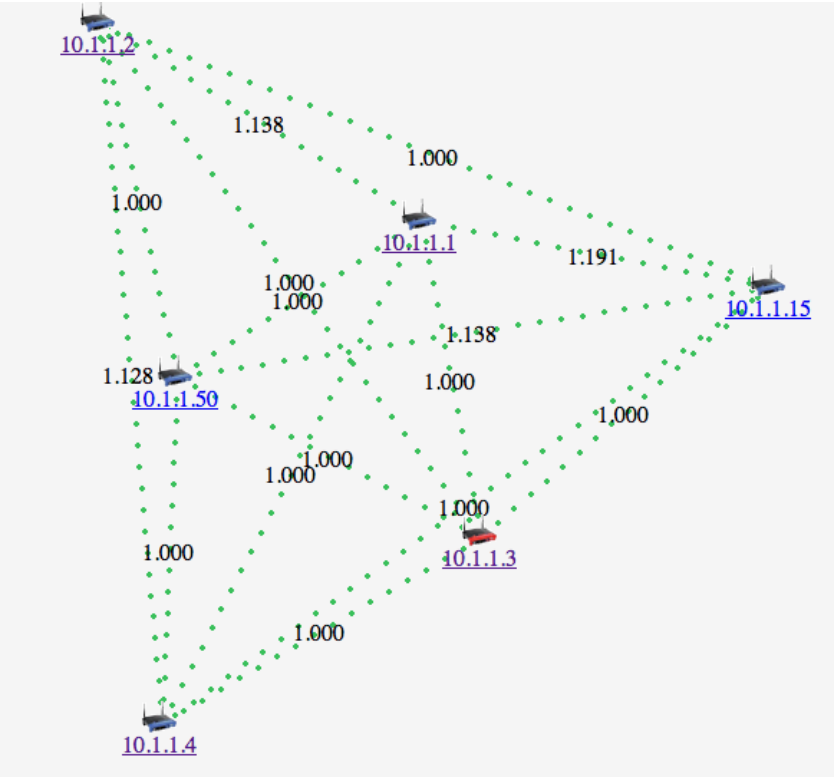
Brži i lak pristup računarskoj mreži je karakteristika mesh tehnologije umrežavanja. Pogodnost ove tehnologije jeste ta što se veoma lako nadograđuje postojeća mrežna infrastruktura. Prednosti ove tehnologije umrežavanja su jednostavnost proširenja, razvoja, nadogradnje i pouzdanost s veoma malim brojem prekida. Mesh mrežu sačinjavaju klijenti (End Devices) i ruteri, odnosno čvorovi za prosleđivanje paketa (Slika 2.1).



Slika 2.1 WMN mreža

Čvorišta u mesh mrežnoj infrastrukturi imaju dve funkcije:

- prva je primanje i slanje podataka za korisnike koji su konektovani na čvor, a
- druga (koja je ujedno i karakteristika mesh tehnologije) je prosleđivanje podataka s drugih čvorova, što znači da neki čvorovi u mreži samo prime podatke i šalju dalje. Na slici 2.2 prikazana je topologija potpune mesh mreže (Fully connected mesh) – svaki čvor je povezan s ostalim čvorovima u mreži.



Slika 2.2 Potpuno konektovana mesh mreža

Mesh mreže mogu se dizajnirati korišćenjem dve tehnike – flooding i routing.

Flooding tehnika se u novijim mesh mrežama ne koristi zato što zahteva velike resurse. Ova tehnika podrazumeva širenje podataka kroz celu mrežu dok se ne stigne do čvora kome su podaci namenjeni.

Ruting tehnikom se podaci optimalno šalju do odredišnog čvora, odnosno klijenta spojenog na odredišni čvor.

Podaci koji se šalju pomoću mesh mrežne tehnologije na svom putu prolaze različitim čvorovima do odredišta i taj proces se naziva skakutanje (hopping). Algoritmi koji se koriste da bi se osigurala mogućnost prenosa podataka se nazivaju *self-healing* algoritmi. Ovi algoritmi testiraju svaki čvor tako što šalju test-signal i na taj način otkrivaju greške kod pojedinih čvorova. Ako se otkrije greška na određenom čvoru, taj čvor se zaobilazi, odnosno menja se putanja prenosa podataka. U ovom slučaju komunikacija je veoma sigurna jer postoje alternativni putevi ili rute od izvora do odredišta.

Mesh mreže se često koriste u bežičnim mrežama (Wireless Mesh Network, WMN), ali i u svim drugim tipovima mreža. Razvile su se 80-ih godina prošlog veka u SAD i korišćene su prvobitno u vojne svrhe, a cena takve mreže u to vreme bila je visoka. Danas je cena opreme za mesh mreže mnogo pristupačnija, pa su ovakve mreže sve prisutnije.

IEEE 802.11 je skup standarda za bežične lokalne mreže (WLAN) računara za kućne i kancelarijske potrebe. Po IEEE 802.11 standard bežični prenos za komunikaciju radi na 2.4, 3.6, i 5GHz. 802.11 porodica se sastoji od niza *half-duplex over-the-air* modulacionih tehnika koje u osnovi rade na istom protokolu. Najpopularnije su definisane standardom IEEE 802.11b i 802.11g koje su nastale na osnovu prvog 802.11-1997 standarda iz 1997 (tabela 2.1), ali s određenim izmenama i dopunama. Takođe, najnovija multi-streaming modulaciona tehnika opisana je standardom 802.11n.

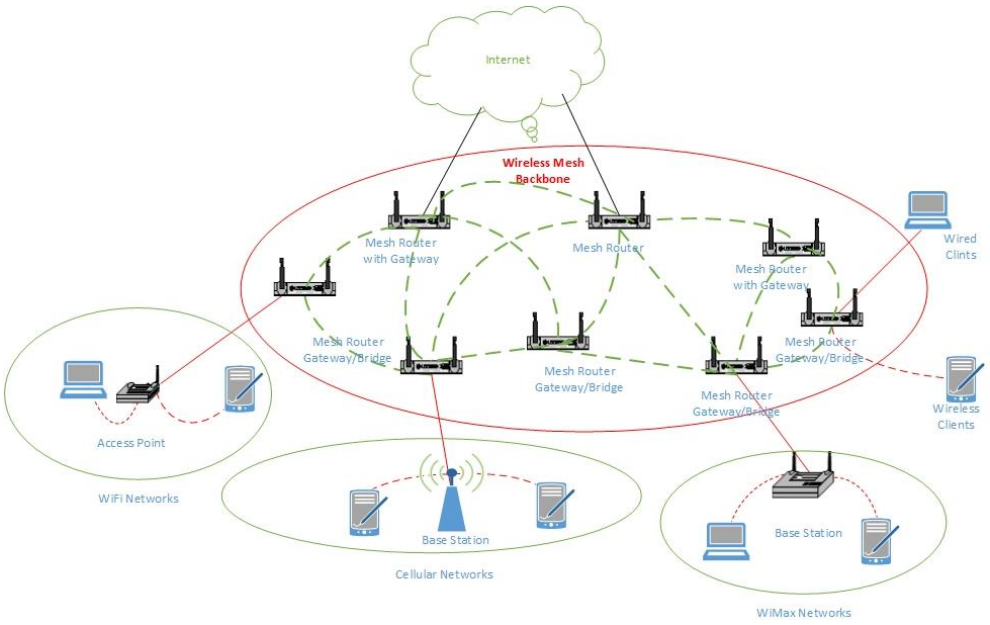
802.11 mrežni standard							
802.11 protokol	**Objavljen**	**Frekvencija (GHz)**	**Propusna moć (MHz)**	**Brzina po strimu (Mbit/s)**	**Modulacija**	**Približan domet u zatvorenom (m)**	**Približan domet na otvorenom (m)**
-	Jun 1997.	2.4	20	1, 2	DSSS, FHSS	20	100
A	Sep. 1999.	5	20	6, 9, 12, 18, 24, 36, 48, 54	OFDM	35	120
		3.7					5000
B	Sep. 1999.	2.4	20	1, 2, 5.5, 11	DSSS	35	140
G	Jun 2003.	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	OFDM, DSSS	38	140
N	Okt. 2009..	2.4	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	OFDM	70	250
		5	40	15, 30, 45, 60, 90, 120, 135, 150		70	250
Ac	Feb. 2014.	5	20	do 87.6	OFDM		
			40	do 200			
			80	do 433.3			
			160	do 866.7			
Ad	Dec. 2012.	2.4/5/60		do 7000			

Tabela 2.1 Mrežni standardi

Bežične mesh mreže formiraju se povezivanjem pristupnih tačaka (Access Point, AP), tako da postoji međusobna veza između svakog AP. Uspostavljanje ovakve veze se prema IEEE 802.11 standardu može realizovati na dva načina: Ad hoc i Distribution System.

U ad hok načinu rada čvorišta (pristupne tačke) komuniciraju podjednako međusobno. *Distribution System* način rada podrazumeva da komunikacijom upravlja AP. U ovom slučaju podaci samo prolaze kroz čvorišta i ona ne upravljaju podacima. Svu kontrolu za upravljanje obavlja AP. Novije mesh mreže obično koriste ad hok način za prenos podataka.

Pre upotrebe IEEE 802.11 standarda za prenos podataka, mesh mreža mora da ispuni nekoliko uslova vezanih za performanse i potrebnih resursa za rad – mora biti određene veličine i imati dovoljno kapaciteta.



Slika 2.3 Različite tehnologije u mesh mrežama

Na slici 2.3 vidimo da je korisnik bilo koje mreže (Wi-Fi, senzorske mreže, WiMAX) na mesh mrežu spojen pomoću AP (Access Point). AP-ovi se s drugim AP-ovima u mreži spajaju pomoću rutera. Ruteri također obezbeđuju i internet celoj mesh mreži.

Kombinovanjem ad hok režima na fizičkom sloju OSI modela i mesh rutinga na mrežnom sloju moguće je stvoriti bežičnu mesh mrežu (Wireless Mesh Networks – WMN) između klijentskih uređaja bez potrebe za centralizovanom pristupnom tačkom ili ruterom.

Po definiciji, ad hok su mreže dinamičkih autonomnih čvorova, međusobno povezanih, koje stvaraju multihop radio-mrežu povezanu u decentralizovanoj strukturi upravljanja (Perkins, 2008). U ad hok mrežama čvorovi su izloženi uticaju interferencije i fedinga. Takođe, linkovi imaju uži propusni opseg u odnosu na fiksne (žičane) mreže. Fizička topologija mreže je promenjiva, što znači da postoji mogućnost da neki čvorovi zbog neispravnosti, gubitka energije ili fizičkog oštećenja mogu da nestanu (izgube signal). Kod ad hok mreža ne postoje pristupne tačke, tako da se poruke prenose između mobilnih čvorova. Ovakvom vezom između čvorova omogućava se veći domet. Veličina i prostranost ad hok mreža varira u zavisnosti od bežične tehnologije za prenos (mobilne mreže, WiFi, WiMax i dr.) i protokola za rutiranje (OLSR, B.A.T.M.A.N, DSDV, AODV i dr.). Čvorovi koji čine ad hok mrežu su osnova za rutirajuću infrastrukturu koja pomoću protokola za rutiranje na trećem nivou OSI modela čini samostalnu mobilnu ili mrežu koja će biti konektovana na neku spoljašnju (Stevanović et al., 2004). Zbog slabljenja u propagaciji, uticaja interferencije i fedinga, kvaliteta linka, potrošnje energije i drugih faktora koji utiču na uređaje, vrlo je teško primeniti zadatu putanju između dva čvora. Da bi se minimizovalo dejstvo ovih faktora,

sposobnost pronalaženja adekvatne putanje je od velikog značaja u ad hok mrežama. Postoji mnogo radova na pronalaženju protokola za ad hok mreže koji može da obezbedi najoptimalniji pronalazak putanje između čvorova pod ovim uslovima.

Neke od karakteristika WMN mreža su:

1. dinamička topologija – čvorovi u mreži su mobilni, mogu da menjaju svoje lokacije, uzrokujući promene topologije mreže;
2. ograničena brzina prenosa – tipične podržane brzine prenosa, od nekoliko Mbps do nekoliko stotina Mbps;
3. ograničeni izvori za napajanje;
4. ograničen domet uređaja;
5. višestruki skokovi između uređaja (multihop) i prosleđivanje informacija povećavaju domet i
6. čvorovi imaju ulogu hostova i rutera.

Primene WMN i bežičnih ad hok mreža su:

1. kontrola saobraćaja na saobraćajnicama, kontrola i upravljanje na parkiralištima s detekcijom slobodnih i zauzetih mesta;
2. kontrola i zaštita u velikim prodavnicama i drugim javnim objektima;
3. dinamična povezanost vrlo velikih oblasti – elementarne nepogode, spasilačke misije, u oblastima gde nema infrastrukture ili je ona razrušena;
4. povezivanje uređaja u WLAN (Wireless Local Area Networks) mreže u kancelarijskim uslovima i
5. monitoring osetljivih instalacija i daljinskog prikupljanja podataka.

U specijalizovanim državnim organima (vojska i policija) za detekciju kretanja, praćenje prisustva opasnih materija, kontrolu povrede zaštićenih prostora, kontrolu i zaštitu pojedinih terorističkih ciljeva itd.

2.1. Evaluacija WMN mreža

Bežične mesh mreže (WMN) (Akyildiz et al., 2005) su komunikacione mreže gde sve različite vrste uređaja deluju kao čvorovi u mreži, povezani međusobno bežičnim linkovima koji formiraju bežičnu multihop mrežu. Tipovi čvorova mogu biti računari, laptopovi, ugrađeni uređaji itd., a broj čvorova može da ide i do nekoliko hiljada. U zavisnosti od mogućnosti čvorova, oni mogu imati ulogu mesh klijenta ili mesh rutera. WMN imaju hijerarhijsku arhitekturu gde mesh ruteri formiraju bežičnu kičmu i mesh klijenti koji su povezani s tom kičmom komuniciraju s drugim mesh klijentima povezanim s istom bežičnom kičmom. WMN su obećavajuće mreže koje mogu biti zamena za postojeće mreže u različitim scenarijima.

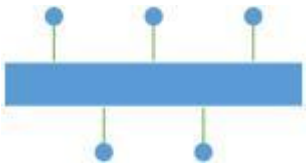
2.2. Podela mreža na osnovu fizičke topologije

Sastavni delovi i način rada računarske mreže mogu se utvrditi na osnovu mrežne topologije. To znači da se osim mesh mreža u praksi koriste i druge topologije. U praksi se mreže najčešće dele na fizičku i logičku topologiju. Mrežna topologija predstavlja fizički raspored uređaja i vezu između hostova, kao i putanju podataka u nekoj mreži.

Point-to-Point mreža se sastoji iz dva čvora i veze između njih. Veza između čvorova može biti stalna (Permanent) i dinamička (Circuit switched, Packet switched). Kod dinamičke veze uspostavlja se komunikacioni kanal pre razmene podataka. Kod Packed switched veze komunikacioni kanal se deli, a podaci se usmeravaju u paketima. Ova vrsta topologije je jeftinija u odnosu na mesh mreže, ali pruža manju brzinu prenosa i nije sigurna (Slika 2.4).

Slika 2.4 Point-to-Point topologija

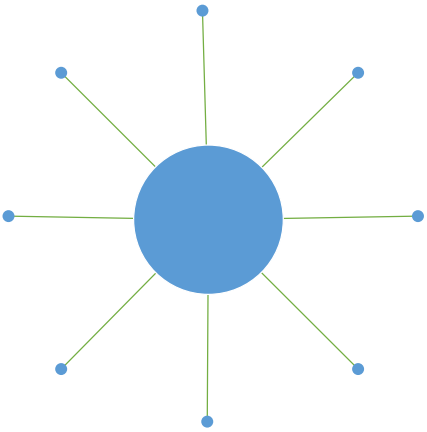
Bus (linijska) mrežna topologija se sastoji od središnjeg vodiča na koji su spojena čvorišta koja komuniciraju (Slika 2.5). Medijum preko koga se odvija komunikacija na oba kraja mora biti terminiran da bi se okončala veza i onemogućila refleksija, tj. odbijanje signala i time smanjile smetnje. Svi podaci šalju se preko središnjeg čvora, a prekid na bilo kom mestu dovodi do prestanka u komunikaciji između svih čvorova. Najčešće se kao medijum za prenos koristi koaksijalni kabl. Bus mrežna topologija jednostavnija je od mesh topologije, ali sigurnost i brzina daju prednost mesh topologiji.



Slika 2.5 Linijska (bus) topologija

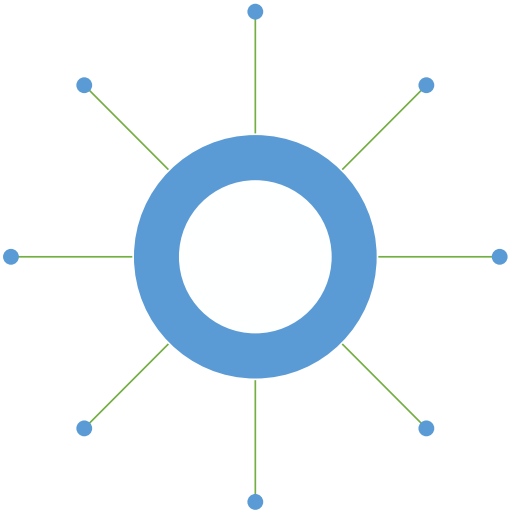
Star (zvezda) mrežna topologija se sastoji od središnjeg čvora na koji su nekim medijumom spojeni ostali čvorovi na mreži. Ulogu središnjeg čvora imaju obično preklopnici (Switch) ili koncentratori (Hub). Ako čvorovi međusobno komuniciraju kroz preklopnik, istovremeno može komunicirati više parova čvorova (Slika 2.6). Glavni nedostatak ovakve mrežne topologije jeste taj što dolazi do prekida na celoj mreži ako središnji čvor prestane da funkcioniše. Prekid rada samo jednog čvorišta ne utiče na komunikaciju ostalih čvorova u mreži što ne doprinosi prekidu rada cele mreže.

Ova topologija je najčešći oblik povezivanja u LAN mrežama. Kao medijum za prenos se koristi UTP kabl. Star mrežna topologija je za implementaciju jeftinija od mesh topologije, ali ako se desi prekid na centralnom čvorištu, prekida se komunikacija na celoj mreži, što kod mesh topologije nije slučaj.



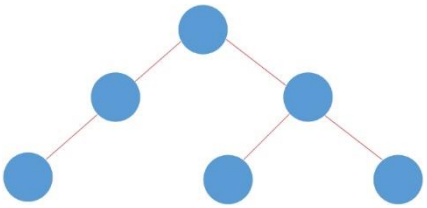
Slika 2.6 Topologija zvezde (star)

Ring (prsten) topologija se sastoji od čvorova koji su povezani sa dva susedna čvora u mreži, kako su prvi i poslednji povezani, stvara se fizički krug ili prsten (slika 2.7). Podaci putuju u krug u jednom smeru od jednog do drugog čvora. Obično se u realizaciji ove topologije pravi i redundantan link između svaka dva čvora – u slučaju kvara na jednom prstenu, koristi se druga redundantna veza. Ova topologija daje veliku brzinu prenosa podataka, ali je zahtevnija od mesh mrežne topologije.



Slika 2.7 Topologija prstena (ring)

Tree (stablo) topologija sastoji se od središnjeg čvora koji je najviši u hijerarhijskom rasporedu čvorova, i na njemu spojenih čvorova koji se nalaze u rangu ispod (Slika 2.8). Čvorovi nižeg sloja mogu imati na sebi spojene čvorove još nižeg sloja. Kao medijum za prenos podataka obično se koristi optički kabl. Tree topologija je najsrodnija mesh topologiji. Ipak, kvar na jednom čvoru može posebno da utiče na mrežu, pogotovu ako se taj čvor nalazi visoko u hijerarhiji.



Slika 2.8 Topologija stablo (Tree)

Logičke topologije najčešće su povezane s načinom na koji se pristupa medijumu za slanje podataka. Oslanja se na primenu komunikacionih protokola, a ne samo na fizički raspored (na primer, logička Ring topologija ne mora da bude i fizička Ring topologija).

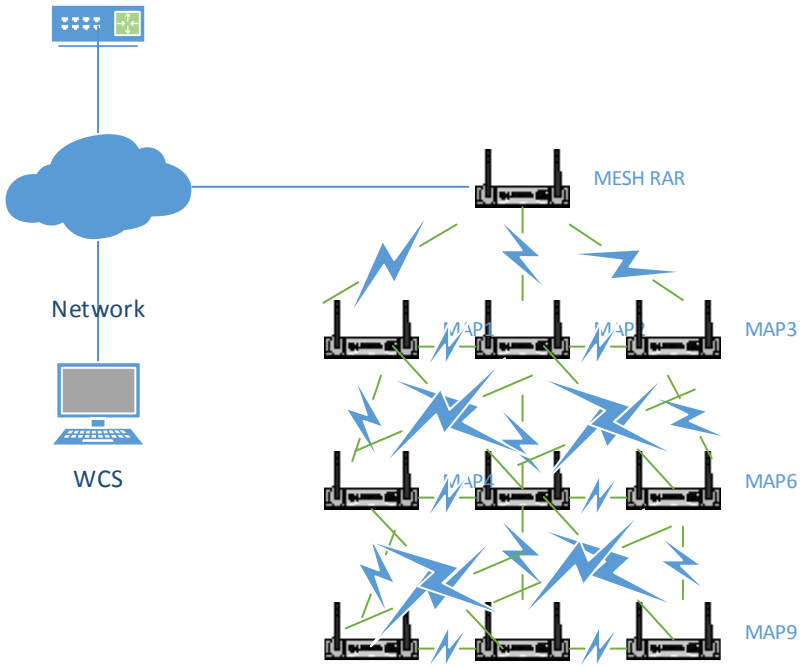
2.3. Topologija WMN mreža

U odnosu na tradicionalne bežične LAN (WLAN) i mreže za mobilnu telefoniju bežična mesh mreža (WMN, Wireless Mesh Network) predstavlja pomak jer se lako nadograđuje i širi. Zbog brojnih prednosti u odnosu na tradicionalne mreže, WMN su budućnost bežičnih mreža. Otpornost na kvarove i jednostavno proširenje su glavne prednosti WMN mreža. Sama topologija, odnosno veliki broj čvorova u mreži čini WMN mreže otpornim na kvarove. Ako postoji kvar na određenom čvoru mreže, pronalazi se drugi put za prenos podataka između korisnika. Instalacija WMN mreža je veoma jednostavna zato što se radi nadogradnja postojećih čvorova.

U zavisnosti od topologije, WMN mreže mogu biti dizajnirane na tri načina – ravni WMN, hijerarhijski WMN i hibridni WMN.

2.3.1. Ravne WMN mreže

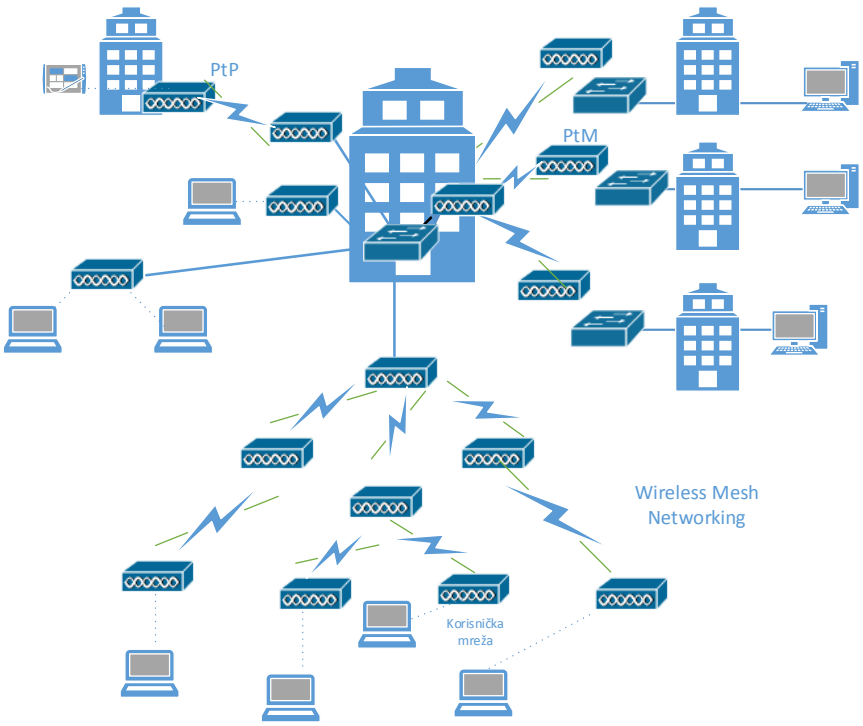
U ravnoj WMN, mreža se stvara od korisničkih uređaja koji se ponašaju i kao domaćini (Host) i kao ruteri. Čvorišta bežičnih korisnika međusobno koordiniraju vezano za ruting, mrežnu konfiguraciju, podršku za servise i prostora za prenos podataka. Ravne WMN mreže su slične centralizovanim bežičnim mrežama (Slika 2.9). Prednost ovakvih mreža je jednostavnost kreiranja, a mana zauzeće mrežnih resursa, što umanjuje brzinu prenosa.



Slika 2.9 Ravna WMN mreža

2.3.2. Hijerarhijske WMN mreže

Osnovna karakteristika hijerarhijskih WMN mreža jeste ta što se takve mreže sastoje iz više nivoa (Slika 2.10). Najniže u hijerarhiji su WMN korisnička čvorišta, odnosno pristupne tačke. Korisnička čvorišta su povezana s osnovom WMN koju čine WMN ruteri. U ovakvoj postavci mesh mreže korisnička čvorišta su zadužena za uspostavljanje veze i prenos podataka. Prednost hijerarhijske arhitekture WMN mreža jeste jeftino postavljanje i velika brzina prenosa u odnosu na ravne WMN mreže, a nedostatak povećan broj čvorova u mreži.



Slika 2.10 Hijerarhijska WMN mreža

2.3.3. Hibridne WMN mreže

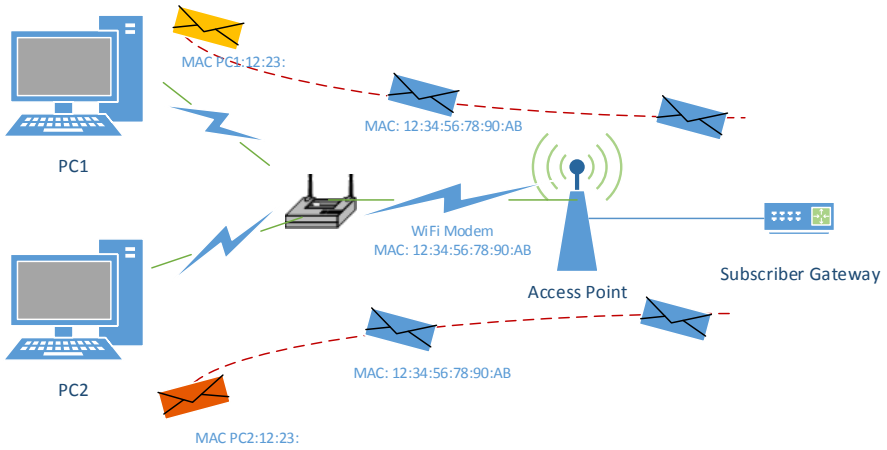
Za hibridne WMN mreže je karakteristično što koriste druge bežične mreže za komunikaciju. Na primer, hibridna WMN mreža može se sastojati iz WiMAX mreže (IEEE 802.16), satelitske mreže (802.11 radios), mreže za mobilnu telefoniju i dr. Ova tehnologija postaje ključna za dalji razvoj WMN mreža.

2.4. Kontrola pristupa u mesh mrežama

Pri dodeli kanala za prenos podataka kroz mesh mrežu koriste se MAC protokoli. Oni mogu da funkcionišu različito, neki dele kanale na potkanale i tako obavljaju paralelnu komunikaciju, dok drugi prenose podatke samo u jednom smeru istovremeno. MAC protokoli takođe mogu da se razlikuju po vrsti enkripcije i enkapsulacije podataka. Da ne bi došlo do greške u prenosu i da bi korisnik imao podršku za protokol koji koristi, važno je da čvorišta u mesh mreži koriste isti MAC protokol. MAC protokoli koji se najčešće koriste u mesh mrežama su:

- Aloha,
- Slotted Aloha,
- CSMA (Carrier Sense Multiple Access),
- CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance),
- IEEE 802.11 MAC protokol,

- IEEE 802.11e QoS MAC protokol.



Slika 2.11 Dodeljivanje potkanala MAC protokola

Primer na slici 2.11 pokazuje prenos podataka dva korisnika (PC1 i PC2). Korisnici su spojeni na pristupnu tačku i pri prenosu koriste MAC protokol, koji dodeljuje potkanale korisnicima od korisničkog uređaja (Wi-Fi Modem/Repeater) do centralnog uređaja (Subscriber Gateway).

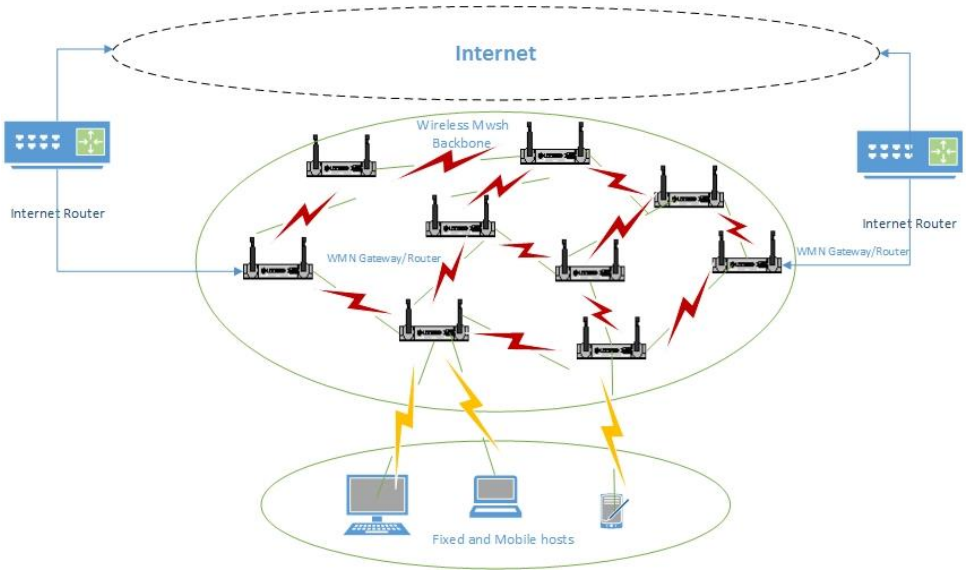
3. WMN PROTOKOLI RUTIRANJA

3.1. Protokoli rutiranja u WMN mrežama

Suština funkcionalnosti mesh mreža jeste u usmeravanju paketa pomoću koga se omogućava primanje i slanje podataka bilo gde u mreži. Protokoli rutiranja (Routing Protocol) u mesh mrežama omogućavaju čvorištima funkcionalnu i optimalnu putanju kroz mrežu. Protokoli rutiranja mogu da proveravaju različite osobine na mreži, na primer zauzetost resursa mreže, greške na čvorovima i dr. omogućavajući pouzdanu i sigurnu komunikaciju u mesh mreži. Postoji veliki broj protokola za upravljanje i usmeravanje paketa kroz mesh mrežu. Najpoznatiji su:

- AODV (Ad hoc On-Demand Distance Vector),
- OORP (Order One Routing Protocol),
- TORA (Temporally-Ordered Routing Algorithm)
- DNVR (Dynamic Nix-Vector Routing),
- OSPF (Open Shortest Path First Routing),
- DSR (Dynamic Source Routing),
- HSLS (Hazy-Sighted Link State),
- IWMP (Infrastructure Wireless mesh Protocol),
- OLSR (Optimized Link State Routing protocol),
- PWRP (Predictive Wireless Routing Protocol)
- DSDV (Destination-Sequenced Distance-Vector Routing),
- B.A.T.M.A.N (Better Approach To Mobile Ad hoc Networking),

I pored toga što postoji veliki broj protokola rutiranja u ad hok mesh mrežama, njihov dizajn se još uvek istražuje kako bi se ubrzao prenos podataka. Na slici 3.1 je dat primer usmeravanja u bežičnoj mesh mreži. Ruting je prikazan crvenim strelicama kojima su povezani WMN usmerivači (WMN Gateway Router). Ruting protokol se pokreće u trenutku kad neko od korisnika želi da prenese podatak kroz mesh mrežu. On određuje put kojim će se podaci kretati kroz čvorišta mesh mreže. Pre samog slanja podataka protokoli rutiranja testiraju dostupnost prenosnog puta, odnosno rute.



Slika 3.1 Ruting u mesh mreži

3.2. Lista ad hok protokola

Ad hok protokoli rutiranja predstavljaju skup pravila ili standard koji određuju na koji način se paketi usmeravaju između računarskih uređaja u mobilnoj ad hok mreži. U ad hok mrežama čvorovi nisu upoznati s topologijom njihove mreže. To znači da novi čvorovi najavljuju svoje prisustvo osluškuju najave i emituju susedima**.** Svaki čvor uči o susedima i kako dolazi do njih.

Podela ad hok protokola rutiranja (Bakht H., 2011):

- Table-driven (proactive) routing
- On-demand (reactive) routing
- Flow-oriented routing
- Hybrid (both proactive and reactive) routing
- Hierarchical routing protocols

3.2.1. Table-driven (proactive) protokoli rutiranja

Ovaj tip ad hok protokola održava liste susednih destinacija i njihovih puteva, na osnovu povremenog distribuiranja tabela rutiranja kroz mrežu. Osnovni nedostaci ovih algoritama su:

- relevantna količina podataka za održavanje i
- spora reakcija na restrukturiranju i neuspešna restrukturiranja.

Primeri **proactive** algoritama su:

- B.A.T.M.A.N. – Better approach to mobile ad hoc networking;
- OLSR Optimized Link State Routing Protocol RFC 3626;

- Babel, a loop-avoidance distance-vector routing protocol RFC 6126;
- Destination Sequence Distance Vector.

3.2.2. On-demand (reactive) protokoli rutiranja

Ovaj tip protokola pronalazi put na zahtev (on-demand) plaveći mrežu paketima Route Request. Osnovni nedostaci ovih algoritama su:

- kašnjenje u pronalaženju puta i
- prekomerne poplave mogu dovesti do zagušenja mreže.

Primeri on-demand algoritama su:

- Admission Control enabled On demand Routing (ACOR)
- Ad hoc On-demand Distance Vector (AODV) (RFC 3561)
- Dynamic Source Routing (RFC 4728)
- Flow State in the Dynamic Source Routing
- Dynamic Manet On-demand Routing (RFC 4728)
- Power-Aware DSR-based

3.2.3. Flow-oriented protokoli rutiranja

Ovaj tip protokola pronalazi put na zahtev (on-demand) praćenjem trenutnih tokova. Osnovni nedostaci ovih algoritama su:

- oslanja se na postojeće saobraćajne entitative da nadoknadi izgubljeno saznanje o rutama. Može da se odnosi na postojeći entitativni saobraćaj kako bi kompenzovao izgubljeno saznanje o rutama i
- potrebno je mnogo vremena za istraživanje novih ruta bez prethodnih saznanja.

Primeri flow-oriented algoritama su:

- IERP (Interzone Routing Protocol/reactive part of the ZRP) i
- RDMAR (Relative-Distance Micro-discovery Ad hoc Routing protocol).

3.2.4. Hybrid (i proactivni i reaktivni) protokoli rutiranja

Ovaj tip protokola kombinuje prednosti proaktivnih i reaktivnih algoritama za rutiranje. Izbor jednog ili drugog metoda zahteva predodređenost za tipične slučajeve. Osnovni nedostaci ovih algoritama su:

- zavisnost od broja drugih aktivnih članova i
- reakcija na zahtev za saobraćajem zavisi od gradijenta obima saobraćaja.

Primer hibridnog protokola:

- ZRP (Zone Routing Protocol) ZRP uses IARP as pro-active and IERP as reactive component.

3.2.5. Hierarchical (hijerarhijski) protokoli rutiranja

Kod ovog tipa protokola izbor reaktivnog ili proaktivnog rutiranja zavisi od hijerarhijskog nivoa u kome se nalazi čvor. Rutiranje se inicijalno zasniva na proaktivnim procedurama, a onda na nižim nivoima na osnovu reaktvnih algoritama prima poplave upita. Izbor jednog ili drugog metoda zahteva odgovarajuća svojstva u zavisnosti od nivoa. Primeri protokola za hijerarhijskim algoritmima su:

- CBRP (Cluster Based Routing Protocol) i
- FSR (Fisheye State Routing protocol).

3.3. Dinamički protokoli rutiranja (DPR)

Postoji nekoliko studija o obavljanju različitih DRP u bežičnim mrežama. Johnson (Johnson et al., 2008) poredi performanse u smislu zaglavlja, protoka, CPU i potrošnje memorije OLSR-a i BMXd-a vršeći merenja na pravom hardveru upotrebom 49-čvorne unutrašnje mesh mreže. (Zakrzewska et al., 2008) i (Ashraf et al., 2007) upoređuju OLSR s AODV, Destination-Sequenced Distance Vector (DSDV) i Dynamic Source Routing protokol (DSR) u smislu troškova ruting zaglavlja, prosečnog kašnjenja i propusne moći; ti radovi proučavaju izvođenje protokola putem simulacije rešetke kao topologije i razmatranjem dela mobilnih čvorova. (Abolhasan et al., 2009) i (Broch et al., 1998) ocenjuju OLSR, DSDV, DSR i AODV na pravoj podlošci za testiranje; međutim, broj čvorova podloške za testiranje je nizak ili se čak i ne spominje, a skalabilnost nije analizirana.

Koliko je poznato, nijedan od prethodnih radova nije istraživao posledice na mrežne protokole rutiranja pri prelasku s Internet protokola verzija 4 (Ipv4) na Internet protokol verziju 6 (Ipv6).

Postoji mnogo mobilnih ad hok mrežnih (MANET) usmeravajućih protokola koji istražuju kombinacije različitih karakteristika (Akyildiz et al., 2005) i (Alotaibi et al., 2012), kao što je izvođenje metrike izvan hop kaunta, unakrsni projekti koji uzimaju metriku iz sloja 2, skalabilnost za velike mreže, ublažavanje poremećaja usluga zbog kvarova na linkovima ili zagušenja itd.

Svim analiziranim DRP je zajedničko:

- rute su postavljene proaktivno
- link susedi se automatski detektuju
- podaci protokola se šalju Ipv6 multikast grupi

- korisnik podataka protokola (UDP) je zaposlen kao transportni sloj protokola
- poruke protokola se grupišu (odnosno, jedan UDP datagram može preneti nekoliko protokolarnih okvira ili poruka).

3.3.1. Babel

Babel je d*estination-sequence distance-vector* (DSDV) ruting protokol naveden u RFC 6126 (Chroboczek, J. 2011). On se zasniva na Bellman-Ford protokolu i koristi izvodljive uslove za odbacivanje ruta za koje nije garantovano da su bez petlji. Babel implementacija analizirana u ovom radu jeste Babeld. Ovu implementaciju je programirao i održava Juliusz Chroboczeka, koji je takođe i autor RFC 6126. Strategija za izračunavanje troškova linka i rute metrike nisu navedeni u RFC 6126. U Babeld-u ovi proračuni su izvedeni upotrebom varijante Očekivane transmisione tačke (Expected Transmission Count – ETX).

Babel karakteriše prilično brza konvergencija jer koristi izazvane ispravke i eksplicitne zahteve za nove ruting informacije. On se obično konverguje odmah pošto se završi merenje kvaliteta linka. Ovo početno rešenje nije optimalno pošto konvergira s jedva zadovoljavajućim setom ruta. Babel polako optimizuje usmeravajuće tabele. U prisustvu teškog gubitka paketa, konvergovanje na optimalni set ruta može trajati znatno duže pošto se ažuriranja rute mogu izgubiti i oporaviti se jedino praćenjem mandatornih periodičnih ažuriranja ruta koje se šalju prilično retko (s podrazumevanim intervalom od 20 sekundi).

U cilju smanjenja troškova protokol Babel omogućava da se izostavi mreža prefiksa kad je više adresa poslato u jednom paketu kao što je opisano (Clausen et al., 2009).

3.3.2. OLSR

Optimizovani link state ruting protokol (OLSR), kao što je navedeno u RFC 3626 (Clausen et al., 2006), predstavlja proaktivni protokol rutiranja koji koristi optimizovanu verziju čistog link state protokola. On je optimizovan u smislu zaglavlja, pošto topološka kontrola poruka ne protiče čisto kroz mrežu već putem multipoint releja (MPR). MPR se biraju na distributivan način, tako da svaki čvor bira mali skup neposrednih suseda da mu budu skup MPR, što znači da svaki dva skoka udaljeni sused može da se dostigne kroz jedan od čvorova u MPR skupu.

Međutim, njegova široka upotreba u postojećim Community Networks (CN) pokazala je da je optimizacija bazirana na MPR nedovoljna kada se suoči s dinamičnim promenama i slabim linkovima koji se pojavljuju u stvarnom životu i samostalnom upoređivanju. Kako bi se to prevazišlo, MPR algoritam je onemogućen u Optimizovanom link stejt usmeravajućem protokolu daemon (OLSRd), trenutno najviše

korišćenim u OLSR implementaciji i analiziranim u ovom radu. U OLSRd, Fish-eye ekstenzija se aktivira podrazumevano kako bi se smanjio prosečni protok saobraćaja zaglavlja i da bi se povećala skalabilnost (Adjih et al., 2004. i Nguyen et al., 2007), a Hop metrička tačka RFC-a je zamenjena ETX-om. Zbog svih ovih i drugih promena OLSRd je postao RFC nekompatibilan već duže vremena.

Trenutno većina postojećih CN koriste OLSRd implementaciju za čitavu mrežu (npr. FreiFunk, FunkFeuer) ili u delovima mreže (npr. Guifi.net, AWMN). Zbog svog prvog većeg angažovanja u komjuniti mrežama 2003, kod se konstantno povećavao i postao veoma stabilan, zreo i buduće dobro rešenje za male i velike mrežne projekte.

OLSR je bio opisan, analiziran i o njemu se dosta diskutovalo u prethodnom radu (Kuppusamy et al., 2011. i Tamilarasan S., 2012). U nastavku se razmatraju najvažniji principi OLSR implementacije upotrebljene za procenu i kako se oni odnose prema protokolu saobraćaja u zaglavlju i kako konvergiraju s vremenom.

OLSR periodično emituje dve vrste poruka:

- HELLO poruke se emituju svake dve sekunde podrazumevano od strane svakog čvora i putuju samo jedan skok. HELLO poruke uglavnom sadrže IP pošaljioca, spisak njegovih suseda i status linka. Oni se koriste za izračunavanje kvaliteta linka između čvorova.
- Topološka kontrola (TC) poruka protiče kroz celu mrežu. U slučaju onemogućenog MPR algoritma, ove poruke se stvaraju u svim čvorovima (u suprotnom, TC poruke protiču selektivno kroz čvorove koje su izabrane kao MPR). TC poruke imaju originator adresu i spisak svojih suseda s odgovarajućim link kvalitetima. TC poruke procesuira svaki čvor za interno izračunavanje pune topološke grafičke mreže, što obezbeđuje osnov za izračunavanje najboljeg sledećeg skoka za svaku destinaciju.

Kao bilo koji link state ruting protokol, OLSR je konceptualno osetljiv na usmeravajuće petlje što rezultira nesinhronizovanim topološkim grafikonima koji su proračunati različitim petljama na forvardujućem putu paketa podataka. Kompromis kod ovog problema dat je protokom TC poruka u manjem intervalu, što omogućava čvorovima da ponovo izračunavaju svoj topološki pregled češće na račun uvećanog protokolaranog saobraćaja u zaglavlju i CPU loudu. Fish-eye ekstenzija je treći način da se ublaži problem. Ona je bazirana na nalazu da se usmeravajuće petlje obično pojavljuju između obližnjih čvorova (zato su čvorovi na udaljenosti od jednog ili dva skoka). Da bi se postigla bolja sinhronizacija topoloških grafikona između obližnjih čvorova dok se omogućava manje frekventna sinhronizacija između udaljenih čvorova, TC poruke teku s različitim TTL vrednostima. Konkretno, sekvenca TTL s aktivnom Fish-eye ekstenzijom u OLSR primeni za naše evaluacije je 2, 8, 2, 16, 2, 8, 2, 225. To znači da

tek svaka parna TC poruka protiče iznad dvoskočnog susedstva. Pošto podrazumevano aktivacija fish-eye ekstenzije kasni 140 sekundi, mora se očekivati prolazno stanje s višim protokolom zaglavlja. Ova zakasnela aktivacija ima za cilj da smanji vreme konvergencije posle pokretanja čvora.

3.3.3. BMX6

B.A.T.M.A.N. eksperimentalna verzija 6 (BMX6) je naslednik B.A.T.M.A.N. eksperimental daemon-a (BMXd) koja se pojavila kao nezavisna grana iz B.A.T.M.A.N. protokola (Neumann et al., 2008) da istraži i testira nove pristupe za usmeravanje i kontekst u širenju mreže. Dizajn i razvoj ove nove verzije je vođen ciljem da se izbori s povećanjem prostora za adrese datog od Ipv6 adresa, da omogući konfiguraciju individualnih čvorova dok pojašnjava rukovanje najavama konfliktnih čvorova (npr. izdvajanje duplikata adrese) i omogući efikasno stanje širenja (stoga je smanjen protokol zaglavlja) kroz strogu razliku između lokalnog i globalnog kao i statičnog i dinamičnog stanja.

BMX6, kao i BMXd aktivno se koriste u tekućim Community Networks (CN) i projektima kao što su guifi.net qMp i Graciasensefils, Freifunk i Lugro-mesh.

BMX6 je *table-driven* ruting protokol za bežične mesh mreže. Kao i svaki drugi *table-driven* protokol rutiranja, njegov cilj je da komponuje put od izvora do odredišta odlučivanjem koji će biti sledeći skok svakom čvoru. BMX6 je *distance-vektor* protokol pošto je informacija kojom svaki čvor upravlja spisak rekorda iz identifikatora čvorova i cena da se dođe do tamo kada se zabere konkretan link je*: <destinacioni čvor, sledeći skok, cena*>. Novina kod BMX6 je mehanizam širenja koji on koristi da propagira ovu informaciju. Protokol širenja je inspirisan ljudskim društvenim mrežama koje su skalabilne jer su ljudi skloni da čuju više o svom susedstvu i apstrahuju i filteruju informacije o drugima. Topološko znanje u čvoru je optimizovano za sebe i njegove susede upotrebom lokalnih kompakt identifikatora za lokalni kompresovani dijalog.

Tokom prolazne faze susedi razmenjuju znanja o svom okruženju, opisima čvorova, linkovima itd. i obezbeđuju informacije o svojim individualnim identifikatorima (Individual IDentifiers – IID) koji identifikuju čvorove na kompaktan način. Sa ovom informacijom, svaki čvor postavlja rečnik tabelu po svom susedu koji prevodi svoje IID vrednosti za globalno jedinstvene i nedvosmislene heševe opisa punog čvora. U stabilnom stanju, svaki čvor ima stanje lokalne informacije u obliku IID-heš rečnika, a stanje globalne informacije kao rečnik heš opisa. Tokom ove faze protokol samo razmenjuje male pakete da bi pratio varijacije link metrike i da prati mrežne promene. Zahvaljujući informacijama raspoređenih tokom prelazne faze, polja ovog periodično razmenljivog usmeravajućeg ažuriranja, koje su obično date 128 bit IPv6 adresom, mogu se zameniti mnogo kraćom IID vrednošću (16 bita) i to rezultira kompresovanom

porukom. Razdvajanje u lokalnom i globalnom stanju takođe se isplati kad se čvor kreće i zato mu se susedstvo menja jer mu je jedino potrebno da ponovo uspostavi IID-heš odnose, a odgovarajuće znanje o parovima između heševa i odgovarajućih opisa i dalje važi.

Kao rezultat, kontrola zaglavlja povećava se kad postoji promena mreže, stabilizujući se potom u nižu vrednost. Stoga, kada se čvor pokrene, moramo da očekujemo značajan skok na samom početku, što rezultira razmenom opisa čvorova i lokalnim IID tabelama. Međutim, pošto se završi inicijalna, prolazna faza, buduće promene na mreži – konektivne varijacije – imaće mnogo manji uticaj na saobraćaj zaglavlja jer veoma malo informacija već razmenjenih tokom početne prolazne faze mora da se ažurira – odnosi se na promenu struje.

Treba uzeti u obzir da bi tipična situacija bila mala promena u mreži, kao što je povezivanje ili isključivanje čvora iz mreže, dok simultano pokretanje svih čvorova u mreži nije uobičajeno. Međutim, sličan efekat može se očekivati u slučaju da dva odvojena mrežna oblaka postanu jedna mreža dolaskom novog linka koji ih povezuje. U ovom slučaju, možemo očekivati visok stepen saobraćaja jer svaki čvor u mreži treba da se upozna sa svim čvorovima u drugom oblaku.

Shodno tome, postoje dve različite vrste poruka na BMX6 što zavisi od njihove prirode: (a) periodične poruke koje su periodično generisane protokolom svakog čvora i (b) povremene poruke, koje se razmenjuju samo kad je to neophodno zbog promene u mreži.

Periodične poruke generisane BMX6 su odgovorne za malo zaglavlje tokom stabilne faze i one su:

- HELLO reklamne (HELLO_ADV) poruke i emituju se svakim HELLO_INTERVAL-om, podrazumevano 0,5 sekundi. One se koriste da izmere kvalitet linka (baziranog na broju primljenih poruka) i da bi se saznalo da li je link živ ili ne.
- Slično tome, izveštaj reklamne (RP_ADV) poruke se periodično emituju HELLO_ADV porukama i stoga svakim HELLO_INTERVAL-om. Oni pružaju pregled primljenih i izgubljenih hello poruka iz svih susednih i srodnih linkova.
- OGM_ADV ili Originator poruke šalju se svakim OGM_INTERVAL-om (podrazumevano na pet sekundi) i propagiran je preko mreže. Oni se koriste da omoguće čvorovima da postanu svesni drugih čvorova koji su dalje od samo jednog skoka i informišu ih o metričkoj stazi do čvora porekla. Međutim, OGM_ADV ne protiču nasumično kroz mrežu, već samo kroz takozvane relevantne linkove. Link je relevantan kad god je neophodno da

dostigne jedan od čvorova u mreži, odnosno on je sledeći skok od najmanje jednog unosa u usmeravajućoj tabeli.

Nasuprot tome, povremene poruke stvaraju vrhunac u saobraćaju kad postoji promena na mreži omogućavajući čvorovima da steknu znanje o svom susedstvu ili čuju pun opis prethodno nepoznatog čvora. Ove poruke su:

- Oglašavanje linka Link advertisement (LINK_ADV) i opcionalno oglašavanje uređaja (DEV_ADV) poruke koje se emituju na zahtev (zbog prijema LINK_REQ_ ili DEV_REQ poruka) da opišu postojanje i dalje osobine mrežnih uređaja i linkova iz perspektive individualnog čvora. Svaka LINK_ADV poruka predstavlja link kako je percipiran (zbog prethodnih primljenih HELLO_ADV) transmitovanjem čvora do jednog od svakog suseda. Red po kome su LINK_ADV poruke sakupljene dalje se koristi kao implicitna referenca za specifičan link čvora kad se stvaraju ili procesuiraju RP_ADV poruke.
- Opisno oglašavanje Description advertisement (DESC_ADV) poruke se razmenjuju između čvorova, pružajući pun opis određenog čvora, koji sadrži detalje kao što su njihove IP adrese, imena domaćina i parametara protokola. Opisne poruke se traže preko DESC_REQ poruka zbog prijema nepoznatog mrežnog opisa.
- Haš oglašavanje (hash advertisement – HASH_ADV) poruke obezbeđuju odnos čvor-specifične IID vrednosti s opisom mreže specifičnog čvora koji se koristi za identifikaciju globalno nedvosmislenog čvora. Sredstvima opisa mreže BMX6 se odnosi na već poznate čvorove bez potrebe za slanjem punog opisa čvora. HASH_ADV poruke su neophodne kad god se primi nepoznata IID referenca ili poruka od nepoznatog čvora.

Ukratko, BMX6 uspeva da smanji svoje zaglavlje pomoću dva različita mehanizma: prvo, on optimizuje saobraćaj koji se periodično transmituje kroz mrežu uspostavljanjem razumevanja između suseda koji koriste kompaktne IID i opisne heševe; drugo, on kontroliše protok poruka analizirajući da li je link relevantan ili nije, i izostavlja nerelevantne linkove pri protoku OGM.

4. SIGURNOST U WMN MREŽAMA

4.1. Pretnje i slabosti u WMN mrežama

Ranjivosti i pretnje u WMN mrežama mogu se podeliti u tri grupe:

- Pretnje i ranjivosti protokola rutiranja (Routing Protocol Threats).
- Pretnje i ranjivosti na nivou korisničkog pristupa mesh mreže (Client Access Threats).
- Fizičke pretnje i ranjivosti (Phisical Security Threats).

Napadi na protokole rutiranja su najčešće vrste napada na WMN mreže. Neke od ovih pretnji zahtevaju ubacivanje paketa podataka u mrežu, kroz propuste u protokolima.

- Crna rupa (Black Hole) – napad kad napadač stvara i prosleđuje pogrešne pakete podataka prilikom čega se stvara novi mesh čvor.
- Siva rupa (Grey Hole) – napad kad napadač ubacuje lažne podatke u mrežu kako bi dobio putanju kojom se podaci kreću, a samim tim i pristup mreži.
- Route error injection – napad kojim se razbija veza između čvorova mesh mreže ubacivanjem poruke koja sadrži grešku o rutiranju paketa (Route error).

U zavisnosti od tehnologije usmeravanja, rizik za ove vrste napada može da bude manji ili veći. Obično su poznati algoritmi rutiranja koji imaju širu primenu u WMN mrežama ranjiviji od algoritama koji su manje poznati. Neke WMN mreže vrše proveru ruting poruka, pa se samim tim smanjuje rizik od napada.

Tabela 4.1 predstavlja različite tipove ranjivosti na različitim slojevima, kao i odbrambene mehanizme u WMN mrežama.

Sloj	Napadi	Odbrambeni mehanizmi
Fizički	Jamming Device tempering	Spread-spectrum, priority messages, lower duty cycle, region mapping, mode change
MAC	Collision	Error-correction code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Mrežni	Spoofed routing information & selective forwarding	Egress filtering, authentication, monitoring
	Sinkhole	Redudancy checking
	Sybil	Authentication, monitoring, redudancy
	Warmhole	Authentication, probing
	Hello Flood	Authentication, packet leashes by using geographic and temporal info
	Ack. Flooding	Authentication, bi-directional link authentication verification
Transportni	Flooding De-synchronization	Client puzzles Authentication
Plikativni	Logic errors Buffer overflow	Application authentication Trusted computing

Tabela 4.1 Različiti tipovi ranjivosti na različitim nivoima OSI

4.2. Vrste napada u WMN mrežama

Ranjivosti u WMN mrežama eksploatisane od strane napadača direktno utiču na performanse mreže. Čvorovi u WMN mreži su zavisni od saradnje s drugim čvorovima mreže. Shodno tome, protokoli MAC nivoa i mrežnog nivoa za ove mreže koriste pretpostavku da čvorovi koji učestvuju nemaju maliciozne namere. U praksi, međutim, neki čvorovi u WMN mreži u obavljanju svog posla ponašaju se sebično ili mogu biti ugroženi od zlonamernih korisnika. Pretpostavljeno poverenje i nedostatak odgovornosti zbog odsustva centralnog administratora čine WMN mreže ranjivim na različite vrste napada na MAC i mrežnom nivou.

4.2.1. Napadi na fizičkom sloju

WMN mreže zahtevaju da pristupne tačke budu izvan fizičkog dometa operatera. Na osnovu toga javlja se mogućnost fizičkih sigurnosnih pretnji pristupnoj tački mesh mreže:

Spoljna implementacija predstavlja veće izazove za fizičku zaštitu sigurnosti. Pristupna mesta mesh van ustanova i uopšteno na mestima koja nisu pod nadzorom mrežnog operatera omogućuju krađu samih uređaja, kao i menjanje konfiguracije.

Žičane mesh pristupne tačke zahtevaju povezivanje preko žičanih medijuma kojeg napadač jednostavno može da ugrozi.

Fizički sloj je odgovoran za izbor frekvencije, detekciju signala, modulaciju i enkripciju podataka. Kao kod svakog radio zasnovanog medijuma, mogućnost ometanja na tom sloju kod WMN mreža je uvek moguć. Ometanje signala je vrsta napada kome su podložni čvorovi mreže – praktično se zauzima frekvencija na kojoj čvorovi imaju komunikaciju. Izvor koji ometa signal može biti toliko snažan da poremeti komunikaciju u mreži. Čak i s manjom jačinom ometajućeg signala, protivnik može potencijalno da dovede do prekida komunikacije u celoj mreži strateškom distribucijom ometajućeg izvora.

4.2.2. Napadi na MAC sloju

Moguće su različite vrste napada na MAC sloju. Neki od najpoznatijih su: pasivno prisluškivanje (passive eavesdropping), ometanje (jamming), lažiranje MAC adresa (MAC address spoofing), ponavljanje (replay), nepravičnost u raspodeli (unfairness in allocation) itd.

Pasivno prisluškivanje (passive eavesdropping): priroda emitovanja prenosa u bežičnim mrežama čini ih sklonim pasivnom prisluškivanju u prenosnom opsegu do komunikacionih čvorova. Multihop bežične mreže kao što su WMN su takođe pogodne za unutrašnje prisluškivanje zbog intermedijalnih skokova pri čemu zlonamerni srednji

čvor može da zadrži kopiju svih podataka koji se prenose bez znanja drugih čvorova u mreži. Iako pasivno prisluškivanje ne utiče direktno na funkcionalnost mreže, to dovodi do kompromisa u integritetu i tajnosti podataka (Khan et al., 2008). Enkripcija podataka pomoću ključeva je generalan način za zaštitu integriteta i tajnosti podataka.

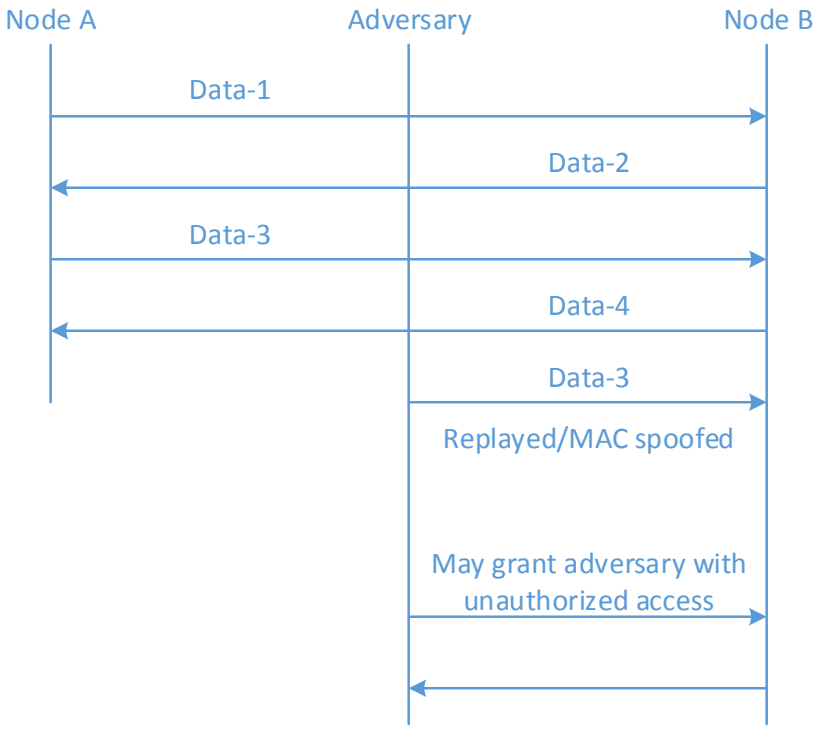
Napad ometanjem linka (Link layer jamming attack): Napadi ometanjem linka su mnogo složeniji napadi od napada na fizičkom sloju. Umesto konstantnog emitovanja slučajnih bitova, napadač može slati regularna zaglavlja MAC okvira na prenosni kanal koji odgovara MAC protokolu koji se koristi u mreži žrtve (Law et al., 2005). Shodno tome, za legitimne čvorove će kanal konstanto biti zauzet za određeno vreme do ponovnog pronalaženja kanala. Ovo dovodi do onemogućavanja servisa legitimnih čvorova i omogućava ometanje očuvanja energije čvora. Pored MAC sloja, ometanje (jamming) se može iskoristiti za eksploatisanje napada na transportnom sloju (Brown et al., 2006). Inteligentno ometanje nije samo prenošenje aktivnosti. Raspoređeni sofisticirani senzori detektuju i identifikuju mrežne aktivnosti žrtve, s posebnim fokusom na semantiku protokola višeg sloja (e.g.,AODV and TCP). Na osnovu zapažanja senzora, napadači mogu da iskoriste predvidljivo vreme prethodno izloženog protokola višeg nivoa i koristeći oflajn analizu sekvenci paketa da maksimalno iskoriste za potencijalno ometanje. Ovi napadi mogu da imaju efekte bez obzira na to da se u mreži koristi enkripciona zaštita kao sto su WEP (Wired Equivalent Privacy) ili WAP (WiFi protocol access). To je moguće zato što senzori koji omogućuju ometanje mogu da prate veličinu, tajming i redosled paketa na **osnovu čega se i napad svodi.**

Namerna kolizija frejmova: kolizija se javlja kada dva čvora pokušaju prenos na istoj frekvenciji (Wood et al., 2002). Kad se frejmovi sudare, oni su odbačeni i treba da se ponovo pošalju. Protivnik može da izazove koliziju strateški u specifičnim paketima kao sto su “acknowledgment” (ACK) kontrolne poruke. Mogući rezultat ovakvog napada je narušavanje protokola komunikacije pomoću kontinuirano prenesenih poruka koje generišu koliziju. Ponovljene kolizije napadač takođe može iskoristiti i one dovode do iscrpljivanja resursa. Na primer, naivna implementacija MAC sloja može da dovede do kontinuirane retransmisije korumpiranih paketa. Ako se retransmisije ne otkriju ranije, energetski nivo čvora brzo bi se iscrpeo. Napadač može da izazove nepravičnost tako što povremeno koristi MAC sloj napade. U tom slučaju protivnik izaziva degradaciju aplikacija koje rade u realnom vremenu na drugim čvorovima i remeti njihov prenos frejmova.

Napad lažiranjem MAC adresa: MAC adrese se u LAN i WLAN mrežama koriste kao jedinstveni identifikator uređaja. Sastoji se iz dva dela dužine 48 bita i izražava se u heksadecimalnom obliku, i to identifikatora proizvođača (24 bita) i identifikatora samog uređaja (24 bita). Današnji MAC protokoli i interfejs kartice ne podržavaju nikakve mere zaštite koje bi sprečile napadača da promeni izvornu MAC adresu uređaja.

Naprotiv, često postoji puna podrška u drajverima proizvođača, što promenu MAC adrese čini veoma lakom. Izmena MAC adrese naziva se MAC obmana (MAC spoofing), i napadaču može poslužiti u različite svrhe. MAC obmana omogućava napadaču da izbegne sisteme detekcije upada (IDS). Današnji administratori mreže često koriste liste MAC adresa za kontrolu pristupa čvoru. Na primer, samo registrovane MAC adrese mogu da se konektuju na pristupnim tačkama. Napadač može lako da prisluškuje mrežu i utvrdi legitimne MAC adrese. Ovo omogućava napadaču da se maskira kao legitimni korisnik neke mreže. Napadač pri tom može ubrizgati veliki broj frejmova što može da izazove uskraćivanje usluga za legitimne čvorove na mreži.

Napad ponavljanja (Replay attack): napad ponavljanja može da pokrene eksterni ili interni čvor u mreži. Eksterni maliciozni čvor može da prisluškuje komunikaciju između čvorova u mreži (Slika 4.1). Ako eksterni čvor prenosi legitimne poruke, u kasnijoj fazi postaje legitimni čvor sa svim pravima pristupa mrežnim resursima. To znači da interni zlonamerni čvor, koji je posrednik između dva čvora, može da napravi kopiju prenetih podataka. Na osnovu toga može da reemituje te podatke da bi obezbedio neovlašćeni pristup mrežnim resursima.



Slika 4.1 Prisluškivanje čvorova u mreži koje vrši eksterni maliciozni čvor

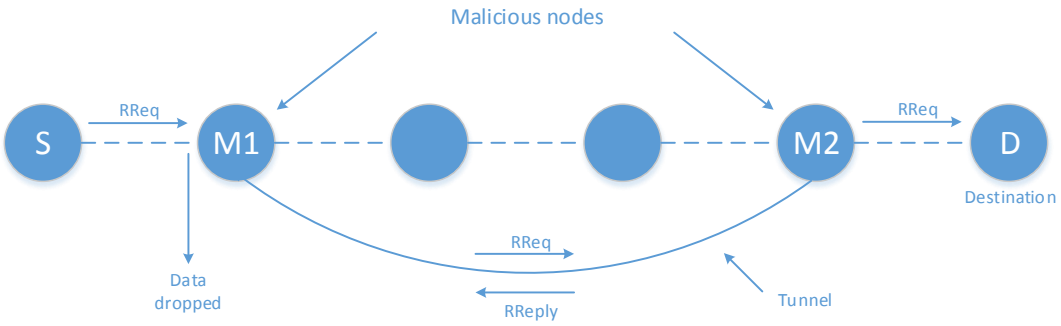
TMTO (Time Memory Trade-Off attack) napadi: za razliku od navedenih napada koji se zasnivaju na ranjivosti samog MAC protokola, TMTO napadi iskorišćavaju

ranjivosti bezbednosnih mehanizama koji su zaduženi. Ovi napadi su veoma efikasni protiv velikog broja kriptografskih rešenja.

4.2.3. Napadi na mrežnom sloju

Napadi na mrežnom sloju mogu se podeliti na napade kontrole i napade na podatke, koji po prirodi mogu biti aktivni ili pasivni. Meta kontrolinih napada je prvenstveno ruting na mrežnom nivou. Cilj napadača jeste da onemoguće razmenu informacija o ruting tabelama između čvorova ili da onemoguće optimalnu putanju. S druge strane, napadi na podatke utiču na funkcionalnost prosleđivanja paketa kroz mrežu. Cilj napadača jeste da onemogući servise za legitimne članove mreže ili da importuje zlonamerne podatke u mrežu.

Napadi na kontrolnom planu: nagli napadi (Hu et al., 2003a) usmereni ka ruting protokolima na zahtev (npr. AODV), među prvim napadima izloženim na mrežnom sloju multihop bežičnih mreža. Nagli napadi koriste mehanizam pronalaženja putanje na zahtev, protokola rutiranja. U ovim protokolima, čvor zahteva put do odredišta poplavom route_request (RREQ) poruka, koje se identifikuju pomoću sekventnog broja. Da bi se sprečila poplava, svaki čvor prosleđuje samo prvu poruku koju dobije, a odbacuje ostale s istim sekventnim brojem. Da bi se izbegla kolizija poruka, protokol definiše određeno vreme odlaganja između route_request zahteva za određeni čvor, i njegovo prosleđivanje ka istom čvoru. Zlonamerni čvor pokreće nagle napade slanjem velikog broja RREQ poruka na određeni čvor pre bilo koje komunikacije čvora od izvora do odredišta. To se lako može ignorisanjem navedenih kašnjenja. Znači, put od izvora do odredišta može da sadrži eliminisani i zlonamerni čvor, koji prosleđivanjem paketa dovodi do DoS napada.

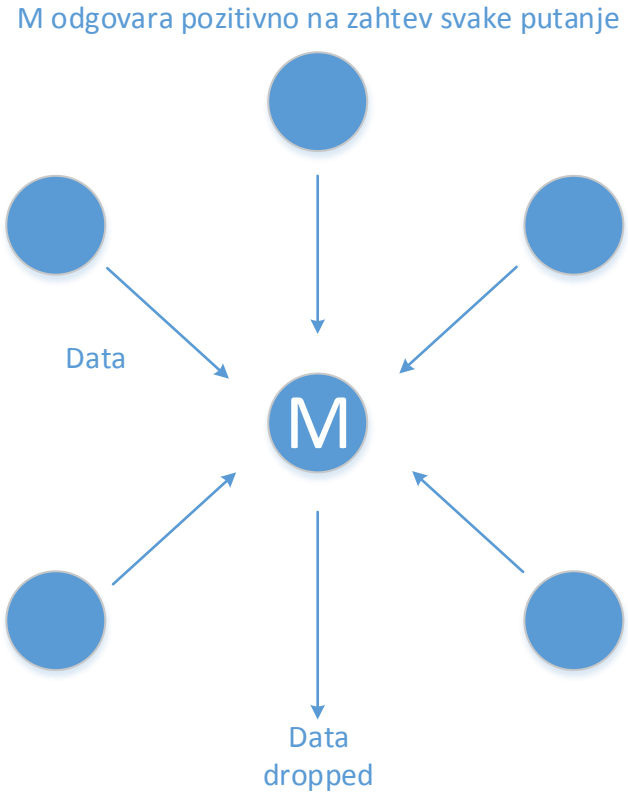


Slika 4.2 Wormhole napad koji su pokrenuli nodovi M1 i M2

Wormhole napad ima sličan cilj, iako koristi drugačiju tehniku (Hu et al., 2003). Tokom ove vrste napada, dva ili više zlonamernih čvorova uspostavljaju zajednički tunel pomoću efikasnog komunikacionog medijuma (Slika 4.2). Tokom faze otkrivanja putanje na zahtev protokola rutiranja, RREQ poruke se prosleđuju između zlonamernih čvorova. Kad su zlonamerni čvorovi uključeni u proces rutiranja, mogu da povuku sve

pakete što bi kao rezultat prouzrokovalo DoS napad, ili da propuštaju selektivno deo protoka da bi izbegli otkrivanje.

Blackhole (Crna rupa) napad (Al-Shurman et al., 2004) je još jedana vrsta napada koja dovodi do uskraćivanje usluga (DoS) u WMN mrežama. On takođe iskorišćava mehanizam otkrivanja putanje na zahtev protokola rutiranja (Slika 4.3). U blackhole napadu zlonamerni čvor uvek odgovara pozitivno na RREQ zahtev, iako u nekim slučajevima neće imati važeći put do odredišta. Sav saobraćaj u susedstvu zlonamernog čvora biće usmeren ka drugom zlonamernom čvoru, koji će izostaviti sve pakete, što dovodi do uskraćivanja usluga. Složeniji oblik je kooperativni blackhole napad gde više zlonamernih čvorova dogovaraju zajedno, što dovodi do potpunog prekida rutiranja i funkcionalnosti paketske mreže. Kooperativni blackhole napadi i mehanizmi za sprečavanje su opisali Ramaswamy et al., 2003.



Slika 4.3 Blackhole napad pokrenut pomoću noda M

Grayhole napad je jedna varijanta blackhole napada (Sen et al., 2007). Prilikom blackhole napada zlonamerni čvorovi crpe sav saobraćaj, što omogućava laku detekciju napada. Tokom grayhole napada protivnik propušta deo paketa selektivno i tom prilikom može biti nezapažen duže vreme. To je zato što se odbacivanje paketa može smatrati kao zagušenje u mreži, što i u ovom slučaju rezultira gubitkom paketa.

Sybil napad je oblik napada gde zlonamerni čvor stvara više identiteta u mreži, a svaki prijavljuje kao legalan čvor (Newsome et al., 2004). Ovaj tip napada prvi put je izložen

u distribuiranim računarskim aplikacijama gde se redundantnost u sistemu eksploatiše stvarajući višestruke identitete i kontroliše značajne resurse sistema. U umrežavanju, usluge kao što su ruting, prosleđivanje paketa i bezbednosni mehanizmi mogu biti poremećene pomoću *Sybil* napada. Ovaj oblik napada utiče na mrežni sloj WMN. Ako zlonamerni čvor stvara više identiteta u mreži, legitimni čvorovi će pretpostaviti da su ovi identiteti legalni i tom prilikom dodaće ove identitete u listu različitih putanja na raspolaganju za određenu destinaciju. Na ovakav način zlonamerni čvor može pokrenuti više vrsta napada koji su prethodno navedeni.

Na mrežnom sloju postoji još nekoliko vrsta napada: poplava upitima za putanju (RREQ flooding attack), petlja odgovora na upit za putanju (route reply loop attack), redirekcija putanje (route re-direction attack) itd. RREQ poplava (RREQ flooding) jedan je od najjednostavnijih napada u kojima zlonamerni čvor pokušava da poplavi celu mrežu RREQ porukama. Kao posledica toga, ovo izaziva veliki broj nepotrebnih komunikacija u mreži, što dovodi do gubitka energije i opadanja propusnog opsega. Petlja odgovora na upit za putanju je vrsta napada gde put prolazi kroz iste čvorove i ponavlja se. Rezultat ovog napada je potrošnja resursa i izolacija odredišnog čvora.

Data plejn napadi (Data plain Attack): ove vrste napada prvenstveno pokreću sebični zlonamerni čvorovi u mreži, koji dovode do onemogućavanja usluge i degradacije legitimnog korisničkog saobraćaja. Najjednostavniji data plejn napad je prisluškivanje. Prisluškivanje je napad na MAC sloju. Ovakvo sebično ponašanje učesnika WMN mreže predstavlja veliki bezbednosni problem jer dostavljanje podataka zavisi od čvorova u WMN mreži. U ovakvoj situaciji teško je napraviti razliku između sebičnog ponašanja zlonamernog čvora i zagušenja u mreži. Značajni mrežni resursi mogu se opteretiti neželjenim paketima, što dovodi do uskraćivanja usluga za legitimni korisnički saobraćaj. Maliciozni čvorovi mogu ubrizgati zlonamerne kontrolne pakete i dovesti do prekida rutiranja.

4.2.4. Napadi na transportnom sloju

Najčešće vrste napada na transportnom nivou kod WMN mreža su poplave (flooding attack) i desinhronizacija (de-synchronization attack). Kad god protokoli održavaju vezu na oba kraja konekcije, postaju ranjivi na memorijsku iscrpljivost napadom poplave. Napadač ponavlja zahtev za povezivanje više puta, što dovodi do iscrpljivanja resursa. U ovom slučaju dodatni legitimni zahtevi će se ignorisati. Desinhronizacija se odnosi na prekid postojeće veze (Wood et. al., 2002). Napadač može, na primer, konstantno da potražuje reemitovanje okvira od domaćina. Takođe, napadač u ovom slučaju može da degradira poruke između domaćina i spreči njihovu komunikaciju jer domaćini gube energiju pokušavajući da isprave greške.

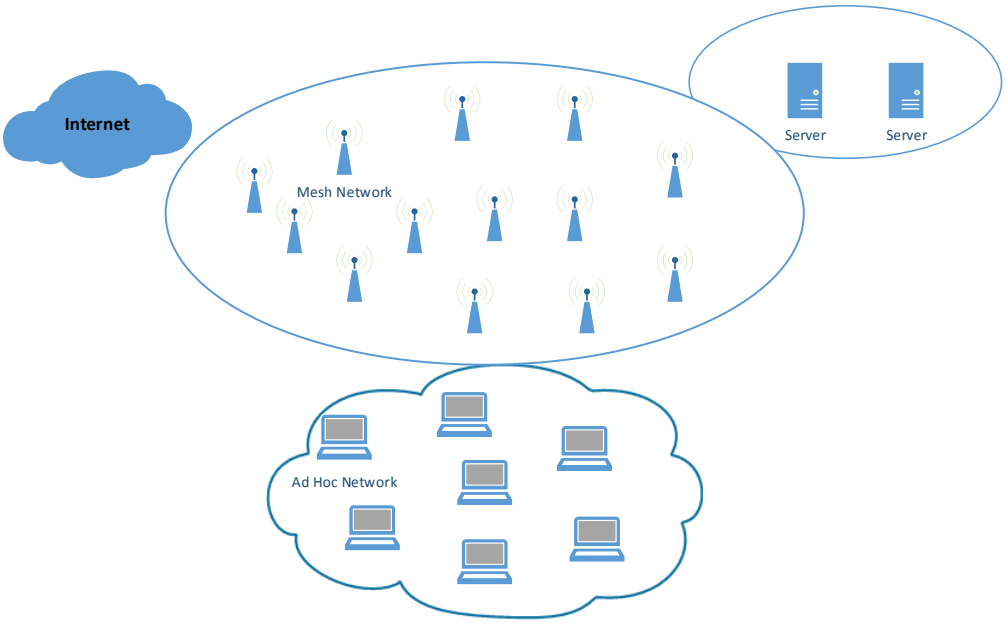
4.3. Pretnje i ranjivosti na nivou korisničkog pristupa WMN mreže

Pretnje na nivou korisničkog pristupa zavise od vrste mesh mreže i strategiji pristupa mrežnim resursima. Mreže koje obezbeđuju besplatan javni pristup ranjive su na napade zasnovane na otvorenoj autentikaciji. Postaje tri vrste napada na nivou korisničkog pristupa:

- Lažiranje mrežne infrastrukture (Spoofing of wireless infrastructure) – napadač koristi tzv. *Evil Twin* ili *Man-In-The-Middle* napad koji omogućuje zajedničku autentikaciju i pristup mreži zajedno s ovlašćenim korisnikom mreže.
- Napad uskraćivanjem usluge (Denial of Service) – napad na IP adrese korisnika, mrežne servise, MAC (Media Access Control) servise da bi uskratio usluge korisnicima mesh mreže.
- Krađa usluge (Theft of Service) – napadač preuzima važeće korisničke podatke. Jedan od načina za zaštitu od ovakve vrste napada jeste da se nakon prve autorizacije sačuva MAC i IP adresa klijenta i nakon toga omogući pristup samo s tih adresa.

4.4. Mehanizmi zaštite u WMN mrežama

Većina mesh mreža je kompatibilna s ostalim mrežnim tehnologijama, što čini sigurnosne mehanizme jednakim. Konvencionalni WLAN sigurnosni mehanizmi (WPA2/802.11i) pružaju standardne metode autentifikacije, kontrole pristupa i enkripcije između korisnika i pristupne tačke (Access Point, AP). Ali, uprkos tome, postoji više vrsta mesh mrežnih arhitektura, koje podrazumevaju različit pristup mreži, a time i različit pristup sigurnosti mesh mreže. U budućnosti će sve mesh mreže imati standardizovane sigurnosne mehanizme zasnovane na 802.11s standardu sigurnosti računarskih mreža. Mehanizmi se odnose na korisnika, ad hok mrežu i centralnu mesh mrežu na koju je ad hok mreža povezana (Slika 4.4).



Slika 4.4 Ad hok mreža povezana na centralnu mesh mrežu

4.5. Korisnička kontrola pristupa

Mesh mrežna infrastruktura, žičana ili bežična korisnicima omogućuje pristup mreži i njenim resursima. U većini 802.11 zasnovanih mreža, korisnici su LAN/WAN klijenti i nemaju mesh mrežnih mogućnosti, pa zato ne mogu da budu posrednici u prenosu podataka. U novije vreme, neki proizvođači (Motorola, PacketHop) pružaju korisnička mesh rešenja. U zavisnosti od vrste mreže, kontrola pristupa korisnika može se menjati. Na primer, Metro-WiFi mreže koriste Layer 3 autentifikacione servise, dok privatne mesh mreže koriste WPA-2 (WiFi Protected Access II) kontrolu pristupa.

4.6. Ad hok sigurnost

Ad hok metod je najčešći način funkcionisanja mesh mreže, što je već opisano u ovom radu. Kao i većina mrežnih tehnologija, ad hok način rada podložan je različitim sigurnosnim ranjivostima. Ove ranjivosti omogućavaju različite tipove napada koji mogu da izazovu nestanak paketa u mreži, dodavanje novih poruka, zauzimanje čvora mreže i dr. Algoritmi rutiranja u mesh mreži najčešće ne specificiraju zaštitu podataka. Iz tog razloga razvija se metoda ad hok sigurnosti, čiji je cilj zaštita podataka na mesh mreži. Metoda se zasniva na tome da je cela mrežna infrastruktura pod administrativnom i sigurnosnom kontrolom jednog entiteta, odnosno administratora. Kod ovakve vrste kontrole sve mesh pristupne tačke pripadaju jednom administrativnom domenu. Nekoliko ključnih tehnologija za bezbednost mesh mreže koje se zasnivaju na 802.11s (Hiertz et al., 2010) standardu:

- Integritet podataka se štiti korišćenjem javnog/privatnog ključa koji sadrži autentikacijske poruke između čvorova. Ovo osigurava svaki neovlašćeni upad u mesh mrežu.
- Autentikacija poruka za usmeravanje korišćenjem digitalnih sertifikata.
- Zaštita podataka pomoću kriptografije, korišćenjem distribuiranih ključeva i digitalnih potpisa.

4.7. Kontrola između WMN pristupnih tačaka

Iako postoje mnoge napredne tehnologije dostupne preko istraživanja ad hok bezbednosti, većina mesh mreža koristi daleko jednostavniji bezbednosni model u odnosu na mesh bezbednosni standard. Najčešća 802.11 komunikacija između čvorova mesh mreže je WDS (Wireless Distribution System) način. WDS način komunikacije podrazumeva enkripciju sa statičkim ključem i enkripciju s dinamičkim ključem.

- Statički ključ enkripcije koristi WEP (Wired Equivalent Privacy) ili AES (Advanced Encryption Standard) algoritme. Ovakav način zaštite je prilično ranjiv, zato je uveden novi način zaštite podataka između čvorova u mesh mreži koji koristi drugu vrstu enkripcije.
- Dinamički ključ enkripcije koji se zasniva na WPA2/802.11i standardu je višestruko sigurniji od statičkog ključa i najčešće se koristi u komercijalnim mesh mrežama.

4.8. Pregled bezbednosnih protokola rutiranja u WMN mrežama

U oblasti bezbednog unicast rutiranja u *multihop* bežičnim mrežama se intenzivno radilo u proteklom periodu. Kao što je rečeno, napadi na protokole rutiranja mogu da se odraze na proces rutiranja ili na podatke prilikom isporuke, odnosno u oba slučaja. Arijadna i SRS (Li, C., et al., 2011) predlažu da se obezbede izvorni *on-demand* protokoli rutiranja pomoću *hop-by-hop* autentifikacione tehnike da bi se sprečila zlonamerna manipulacija paketa prilikom procesa otkrivanja putanje. SAODV, SEAD i Aran (Li, C., et al., 2011) predlažu da se *on-Demand Distance Vector* protokol rutiranja obezbedi korišćenjem jednosmernog lanca (one-way hash chains), čime bi se obezbedilo propagiranje broja hopova. Autori Papadimitratos et al., 2003, predlažu bezbedni *Link State* ruting protokol koji omogućava ispravnost veze pomoću digitalnih potpisa u jednom smeru lanaca. Da bi obezbedili ispravnu isporuku podataka, Marti et al., 2000 predlažu *watchdog* i *pathrater* tehnike za detekciju malicioznih čvorova kontrolom ispravnosti paketa svakog čvora. SMT (Papadimitratos et al., 2003) i Arijadna (Hu et al., 2002) koristi *multihop* rutiranje da bi sprečili zlonamerne čvorove od selektivnog odbacivanja podataka. ODSBR obezbeđuje otpornost na *vizantijske*

napade detekcijom zlonamernih veza zasnovanih na *end-to-end* potvrdi, na osnovu tehnike povratnih informacija. U HWMP (Hybrid Wireless Mesh Protocol) (Bahr, 2006; Bahr, 2007), *on-demand* čvor omogućava komunikaciju dve mesh tačke (MP) koristeći *peer-to-peer* putanje. Ovaj model se pre svega koristi u slučaju da čvorovi menjaju okruženje i nedostatka glavne MP konfiguracije. HWMP je podložan brojnim napadima kao što su poplave RREQ upitima RREP ruting petljom, preusmeravanje putanje, napad tunelovanjem itd. (Li et al., 2011). LHAP (Lightweight Hop-by-Hop Authentication Protocol) (Zhu et al., 2006) je jednostavan, transparentan autentifikacioni protokol za bežične ad hok mreže. Koristi TESLA metodu (Perrig et al., 2000) da se održi odnos poverenja između čvorova, što nije realno zbog odloženog perioda otkrivanja ključa TESLA metodom.

Za razliku od bezbednosti unikast rutiranja, proučavanje bezbednosnih problema multikast rutiranja u bežičnim mrežama je oskudno. Izuzetak su radovi (Roi et al, 2005) i BSMR (Byzantine-Resilient Secure Multicast Routing) (Curtmola et al., 2009). Roj predlaže autentifikacioni okvir koji sprečava napade na multikast protokole, MAODV (Roier et al., 2000), dok BSMR (Curtmola et al., 2009) dopunjuje rad u (Roy et al, 2005) i predstavlja tehnike zasnovane na merenju koje otkrivaju insajder napade u multicast zasnovanim protokolima.

Ključno je da sva gorenavedena istraživanja bezbednosti unicast ili multicast rutiranja podrazumevaju protokole rutiranja koji koriste samo osnovne metrike rutiranja, kao što su *hop-count* i latentnost. Nijedan od njih ne opisuje protokole rutiranja koji uključuju zahtevnije metrike, koji su se pokazali kao kritični za postizanje visokih performansi u bežičnim mrežama. Naprotiv, mnogi od njih čak uklanjaju važnost optimizacije i performansi u postojećim protokolima kako bi se sprečili napadi na bezbednost. Postoji nekoliko studija (Li, C., et al., 2011) koje se bave sigurnosti QoS rutiranja u bežičnim mrežama. Međutim, one zahtevaju jake pretpostavke, kao što su simetrične veze, pravilno ocenjivanje poverenja čvorova, sposobnost da se tačno utvrdi metrika linka uprkos napadima itd. Pored toga, nijedna od njih ne analizira napade u fazi dostavljanja podataka.

Rad predstavljen u Dong, 2009. je prvi takve vrste koji obuhvata i visoke performanse i bezbednost kao ciljeve u multicast rutiranju i razmatra napade u oba slučaja, u uspostavljanju i fazi isporuke podataka. Kao što je pomenuto, bežične mreže su takođe podložne napadima kao što su "rushing*"* i "vormhole*"* napadi. RAP (Hu et al., 2003a) sprečava "rushing" napad koji očekaju poplavu zahteva i zatim nasumično biraju jedan zahtev napred, i ne prosleđuju uvek samo prvi.

U nastavku su predstavljeni neki od poznatih sigurnosnih protokola za rutiranje u WMN mrežama. Ovi protokoli su u stvari ekstenzije osnovnih protokola za rutiranje, kao što

su AODV, DSR itd. i oni koriste kriptografske mehanizme za autentifikaciju, integritet poruke i poverljivost poruke.

4.8.1. Authenticated Routing for Ad Hoc Networks (ARAN)

ARAN protokol jeste *on-demand* protokol za rutiranje koji omogućava kriptografske sertifikate da bi obezbedio sigurnost rutiranja. Ovaj protokol vodi računa o *autentičnosti, integritetu poruke i neporecivosti,* ali podrazumeva manju bezbednosnu koordinaciju između čvorova.

Tokom procesa otkrivanja putanje kod ARAN, izvorni čvor oglašava *route_request* (RREQ) pakete. Destinacioni čvor, po prijemu RREQ paketa, reaguje unikasting povratnim odgovorom koji se zove *route_reply* (RREP) paket. ARAN protokol koristi kriptografski preliminarni proces sertifikacije, propraćen *end-to-end* procesom na relaciji potvrde identiteta, čime se postiže bezbedno uspostavljanje putanje. Protokol zahteva upotrebu pouzdanih sertifikat servera T, čiji je javni ključ poznat svim čvorovima u mreže. *End-to-end* potvrda identiteta postiže se tako što se izvor uveri da li je prava destinacija zaista postignuta. Izvor veruje odredištu da izabere povratni put. Protokol je ukratko objašnjen u nastavku.

Izdavanje sertifikata: ARAN koristi ovlašćenim pouzdani server čiji javni ključ je poznat svim legitimnim čvorovima u mreži. Implementacija protokola podrazumeva da su generisani ključevi apriori od servera i distribuirani svim čvorovima u mreži**.** To ne precizira nikakav poseban algoritam za distribuciju ključeva. Na ulazak u mrežu, svaki čvor dobija sertifikat od pouzdanih servera. Sertifikat dobijen od čvora iz pouzdanih servera T ima sledeću formulu:

$$T \rightarrow A : cert_A = [IP_A, K_{A+}, t, e]K_{T-}$$

U (1), IP_A, K_{A+}, *t*, *e* i K_T- predstavljaju IP adresu čvora, javni ključ čvora, vreme izrade sertifikata, vreme isteka sertifikata i privatnog ključa od servera, respektivno.

End-to-end autentifikacioni put: glavni cilj *end-to-end* procesa provere identiteta putanje jeste da se osigura dobijanje prave destinacije od izvornog čvora. Izvorni čvor S emituje RREQ (tj. otkrivanje putanje) paket namenjen odredišnom čvoru D. RREQ paket sadrži identifikator paketa (proces otkrivanja putanje (RDP)), IP adresu odredišta (IP_D), sertifikat izvornog čvora S ($Cert_S$), trenutno vreme (t) i trenutnu vrednost N_S. Proces se može označiti kao u (2), gde je, K_S privatni ključ izvornog čvora S.

$$S \rightarrow broadcasts := [RDP, IP_D, Cert_S, N_S, t]K_{S-}$$

Kad god izvor šalje zahtev za otkrivanje putanje, ona povećava vrednost za trenutno stanje. Vrednost za trenutno stanje je brojač koji se koristi zajedno s vremenskom markicom u cilju njegove lakše promene. Kad čvor dobije RDP paket od izvora s većom

vrednošću, vrednost za trenutno stanje koje je pri prethodnom primanju RDP paketa od istog izvora čvora pravi zapis od suseda od koga dobija paket, enkriptuje pakete sa sopstvenim sertifikatom i emituje ga dalje. Proces je ovako predstavljen (3):

$$A \rightarrow broadcasts := [[RDP, IP_D, Cert_S, N_S, t]K_{S^-}]K_{A^-}, Cert_A$$

Središnji čvor B na prijemu RDP paketa uklanja sertifikat svog suseda, ubacuje svoj i prosleđuje paket dalje. Destinacioni čvor, na prijemu RDP paketa, proverava sertifikat a zatim odgovara s *route reply* (REP). Destinacija unicasts REP paketa do izvornog čvora kroz obrnutom putu:

$$D \rightarrow X := [REP, IP_S, Cert_D, N_S, t]K_{D^-}$$

U (4), čvor X je sused odredišnog čvora D, koja je prvobitno prosledio RDP paket do čvora D. REP paket sledi istu proceduru na obrnutom putu kao procedura za *route-discovery* paket. Poruka o grešci generiše se ako vreme ili za aktuelna vrednost ne odgovara zahtevima, ili ako je sertifikat pogrešan. Poruka o grešci liči na ostale pakete, osim što je identifikator paketa zamenjen ERR porukom.

ARAN je robustan protokol i efikasan je na napade poput neovlašćenog učešća, lažne signalizacije ruta, napad usmeravanjem poruka, promene poruka rutiranja, obezbeđivanje najkraće putanje i ponovnih napada.

Međutim, pošto ARAN koristi *publickey* kriptografiju za proveru identiteta, posebno je osetljiv na DoS napade na osnovu poplave mreže s lažnim kontrolnim paketima za koje su potrebne provere potpisa. Sve dok čvor ne može da proveri potpis potrebnom brzinom, napadač može da natera čvor da odbaci neki deo kontrolnih paketa koje prima.

4.8.2. Secure Efficient Ad Hoc Distance Vector (SEAD)

Secure Efficient Ad Hoc Distance Vector (SEAD) (Li, C., et al., 2011) je proaktivan i bezbedan ad hok protokol za rutiranje zasnovan na *destination-sequenced distance vector* (DSDV) protokolu rutiranja (Perkins et al., 1994). Ovaj protokol je pre svega dizajniran da prevaziđe napade bezbednosti kao što je DoS i napade na korišćenje resursa. Protokol koristi funkciju u jednom smeru i ne podrazumeva nikakvu asimetričnu kriptografsku operaciju. Osnovna ideja SEAD protokola jeste da potvrdi autentičnost broja sekvence i metriku tabele za rutiranje poruke koristeći elemente heš lanca. Prijemnik takođe potvrđuje verodostojnost pošiljaoca i garantuje da usmeravanje informacija potiče iz pravilnog čvora. Izvor svake *routing update* poruke je takođe potvrđen identitet kako bi se sprečilo stvaranje ruting petlje koju napadač pokreće napadom lažiranja, odnosno lažnim predstavljanjem. U nastavku je izložen opis baze DSDV protokola.

Distance Vector ruting: *Distance Vector* protokoli rutiranja pripadaju kategoriji *table-driven* protokola. Svaki čvor održava ruting tabelu sa spiskom svih poznatih puteva do raznih destinacija čvorova u mreži. Metrika koja se koristi za rutiranje je rastojanje izraženo u hopovima (hop-count). Ruting tabele se periodično ažuriraju razmenom informacija o rutiranju. Alternativa ovom pristupu je aktiviranje ispravke, koju svaki čvor emituje ukoliko je routing table bude promenjena. DSDV protokol za ad hok bežične mreže i WMN koristi broj sekvence oznake za sprečavanje formiranje petlji, da bi se sprečio *count-to-infinity* problem, za bržu konvergenciju. Kad se novi *update* paket primi za destinaciju, čvor ažurira odgovarajuću stavku u tabeli rutiranja samo ako je redni broj na primljenom veći nego snimljeni unos u ruting tabeli. Ako su redni broj i ranije zabeležen broj sekvence jednaki, i ako postoji nova vrednost za metriku (rastojanje u broju skokova), i u ovom slučaju se vrši ispravka. Inače bi update paket bio odbačen. DSDV koristi *triggered* ažuriranje (za važne routing promene) pored redovnog periodičnog ažuriranja.

Jednosmerna haš funkcija: SEAD koristi autentifikaciju *update* poruka koje dobija od nezlonamernih čvorova i zlonamernih čvorova. Ovo smanjuje šanse za napade na resurse izazvanih od zlonamernih čvorova. SEAD koristi jednosmernu haš funkciju za proveru *update* poruka.

Postojeći DSDV-DK protokol rutiranja zasniva se na principima jednosmerne haš funkcije. SEAD protokol podrazumeva gornju granicu metrike. Na primer, ukoliko se koristi metrička udaljenost, onda gornju granicu vrednosti *m - 1* definiše maksimalni prečnik (maksimum dužina svih puteva između para čvorova) kod ad hok bežičnih mreža ili WMN. Dakle, protokol za rutiranje pretpostavlja da postoji ruta dužine veće od *m* hopova između bilo koja dva čvora.

Ako se niz vrednosti za čvor izračunava korišćenjem hash funkcije *H* i data je sa (h_1, h_2, ... h_n), gde je *n* deljiv sa *m*, onda je za table unosa sa brojem sekvence *i* $k = \frac{k}{m} - i$.

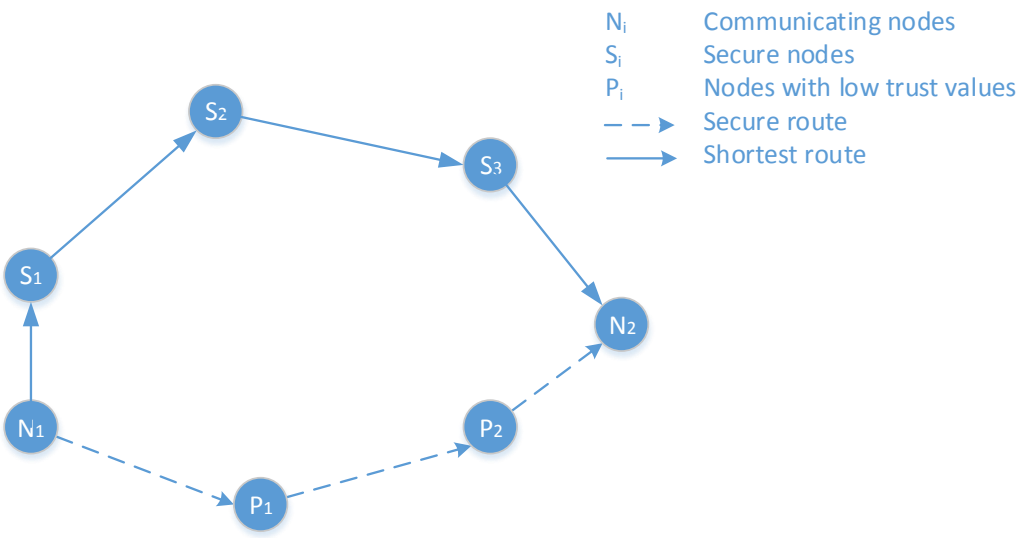
Ako je metrika *J* rastojanje za tu ruting tabelu, $0 \leq j \leq m-1$, vrednost h_{km+j} se koristi za proveru ruting identiteta ažuriranja, za taj broj sekvence *i* metriku *j*. Uvek kada se pošalje *update* poruka, čvor dodaje vrednost koja se koristi za proveru identiteta, ukupno. Ako se koristi autentifikacija vrednost je h_{km+j}, onda napadač koji pokušava da modifikuje ovu vrednost može učiniti samo ako je ona zna h_{km+j-1}. Pošto je jednosmerni lanac h_{km+j-1}, ovo postaje nemoguće. Srednji čvor, na prijemu autorizovanog update-a, izračunava novu haš vrednost zasnovanu na prethodnom update h_{km+j-1}, vrednosti metrike i broja sekvence.

SEAD izbegava ruting petlje, osim ako petlja sadrži više od jednog napadača. Ovaj protokol može se lako implementirati s manjim izmenama od DSDV protokola. Upotreba jednosmernog lanca za proveru identiteta u velikoj meri smanjuje

kompleksnost izračunavanja. Osim toga, protokol je otporan na više nekoordisanih napada. Glavni nedostatak je potreba entiteta od poverenja u mreži za distribucijom i održavanjem elemenata verifikacije svakog čvora jer verifikacija elemenata heš lanca je odvojena od entiteta poverenja. To vodi ka jednoj tački neuspeha u protokolu. Ako je entitet poverenja ugrožen, cela mreža postaje ranjiva. Osim toga, protokol je ranjiv i u situacijama gde napadač koristi iste metričke vrednosti i broj sekvence koje se koriste u prethodnom *update-u* poruke i šalje novi *rutingupdate*.

4.8.3. Security-Aware Ad Hoc Routing (SAR)

Security-Aware Ad Hoc Routing (SAR) protokol koristi bezbednosni ključ u pronalaženju puta i obezbeđivanju okvira za sprovođenje i merenje atributa bezbednosti metrike. Ovaj okvir omogućava korišćenje različitih nivoa bezbednosti za različite aplikacije koje koriste SAR protokol za rutiranje. U WMN mrežama, komunikacija između dva krajnja čvora ili više čvorova zasniva se na činjenici da krajnji čvorovi imaju *trust* vezu sa srednjim čvorovima. SAR definiše nivo poverenja kao mernu jedinicu za rutiranje i kao jedan od atributa za bezbednost koji treba uzeti u obzir. U SAR protokolu, bezbednost metrike je ugrađena u RREQ paket i ponašanje prosleđivanja se sprovodi u odnosu na RREQ pakete. Središnji čvorovi dobijaju RREQ pakete s određenom bezbednosnom metrikom ili nivoom poverenja. Protokol obezbeđuje da čvor prosledi paket ako on može da obezbedi potrebnu sigurnost, ima potrebnu dozvolu ili nivo poverenja. U slučaju da čvor ne može da obezbedi potrebnu sigurnost, RREQ se poništava. Ako je end-to-end put s propisanim sigurnosnim atributima, odgovarajuće izmene RREQ se šalju iz središnjeg do odredišnog čvora. Protokol rutiranja na osnovu nivoa poverenja objašnjava Slika 4.5.



Slika 4.5 Korišćenje metrike poverenja (trust) za čvorove u rutiranju

Kao što je prikazano na slici 6, postoje dve staze između čvorova N1 i N2 koji žele da komuniciraju jedni sa drugima. Jedan od tih puteva je kraći i prolazi kroz privatne čvorove (P1 i P2), čiji je nivo poverenja nizak. Dakle, protokol bira duži put, ali sigurniji i prolazi kroz bezbedne čvorove I1, I2, i I3.

SAR protokol može se objasniti korišćenjem nekog od tradicionalnih protokola za rutiranje. U ovom odeljku, SAR protokol je objašnjen koristeći AODV protokol (Perkins et al., 1999). U AODV protokolu, izvorni čvor emituje *route_request* (RREQ) paket svojim susedima. Srednji čvor, na primanje paketa RREK, prosleđuje dalje, ako nema put do odredišta. U suprotnom, *route_reply* inicira (RREP) pakete nazad na izvornom čvoru koristeći obrnuti put od RREQ paketa. U SAR, određeni nivo bezbednosti uključen je u mehanizam za prosleđivanje (*packet-forward*). Ovde svaki paket ima nivo bezbednosti koji je određen metodom broja obračuna. Svaki čvor posrednik takođe ima određeni stepen sigurnosti. Na prijemu paketa, srednji čvor je takođe povezan s određenim nivoom sigurnosti. Na prijemu paketa, srednji čvor poredi svoj nivo sigurnost s tim definisanim paketom. Ako je nivo bezbednosti čvora manja od nivoa bezbednosti paketa, RREQ paketi se jednostavno odbacuju. Ako je veća, čvor se smatra sigurnim i ima dozvolu da prosledi paket, pored toga što je u mogućnosti da ispita. Ukoliko su nivo bezbednosti srednjeg čvora i primljenog paketa jednaki, onda srednji čvor neće moći da ispita paket (koji može da se obezbedi korišćenjem odgovarajućeg autentifikacionog mehanizma) i samo dalje prosleđuje paket.

Čvorovi na jednakom nivou poverenja distribuiraju zajednički ključ međusobno i s tim čvorovima imaju viši nivo poverenja. Dakle, hijerarhijski nivo bezbednosti u ovakvom slučaju može se održavati. Ovo obezbeđuje da se šifrovani paketi mogu dešifrovati (korišćenjem zajedničkog ključa) samo čvorovi istog ili višeg nivoa bezbednosti u odnosu na nivo bezbednosti paketa. Različiti nivoi poverenja mogu se definisati pomoću brojnih kalkulacija u odnosu na nivo zaštite koji je potreban. Postoji nekoliko metoda. Od aktuelnosti, u cilju isporuke paketa, autentičnost, ovlašćenje, integritet, poverljivost, a ne opovrgavanje su samo neki od karakteristika protokola rutiranja, koje omogućavaju da se definiše nivo poverenja za čvorove i pakete na osnovu broja takvih karakteristika koje se mogu uzeti u obzir.

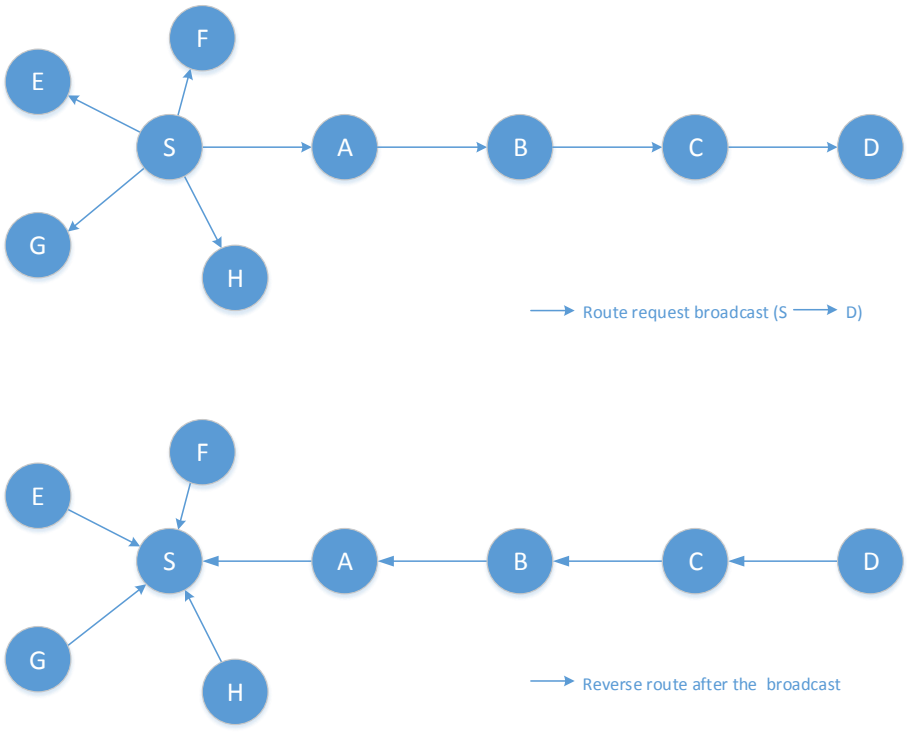
SAR protokol može se lako ugraditi u tradicionalnim protokolima za rutiranje za ad hok bežične mreže i WMN. SAR protokol omogućava aplikaciji da izabere nivo bezbednosti koji ona zahteva. Međutim, protokol zahteva različite ključeve za različite nivoe sigurnosti. Ovde postoji tendencija povećanja broja potrebnih ključeva ukoliko se nivo bezbednosti povećava.

4.8.4. Secure ad hoc on-demand Distance Vector (SAODV)

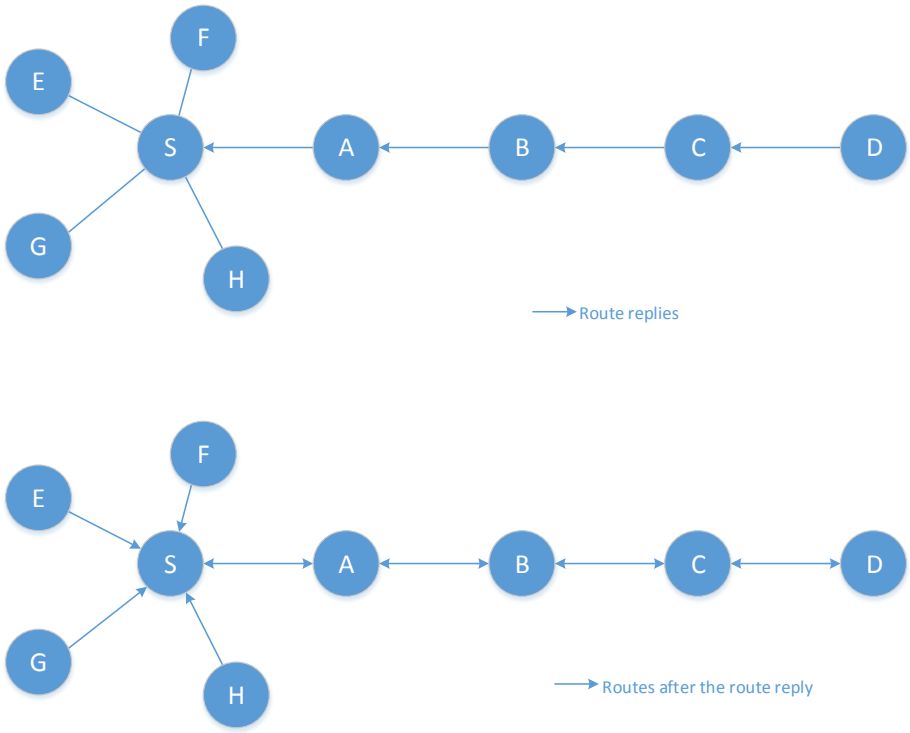
U ovom odeljku će biti opisana AODV verzija protokola gde se razmatraju neke dobro poznate ranjivosti protokola rutiranja. Pre predstavljanja bezbedne verzije (SAODV), kratko je opisana osnova AODV protokola.

***Ad hoc on-demand Distance Vector* (AODV) protokol za rutiranje** je reaktivni protokol za rutiranje (Perkins et al., 2003) za MANETs i WMN mreže koji obezbeđuje puteve samo između čvorova koji treba da komuniciraju. Ruting poruke ne sadrže informacije o celoj ruting putanji, već samo izvor i odredište. Dakle, ruting poruke nisu uvećane. Koriste odredišne sekventne vrednosti da otkriju najnovije putanje, što omogućava izbegavanje petlje.

Kad god čvor treba da pošalje paket na odredište za koju je putanja "dovoljno nova" (tj. važi odredište sa sekvencom čija je vrednost broj barem kao onaj koji se nalazi u svakom RREQ koji je čvor dobio za tu destinaciju), on emituje RREQ poruku svojim susedima. Svaki čvor koji prima poruku postavlja obrnuti put ka začetniku RREQ, ako nema "novije putanje" (Slika 4.6). Kad odredište (ili srednji čvor koji ima "novi" put do odredišta) prima RREQ, on odgovara slanjem RREP. Važno je samo da je promenljiva informacija u RREKQ i u RREP hop-count (koji se monotono povećava na svakom skoku). RREP je *unicast* povratni paket u odnos na začetnika RREQ (Slika 4.7).



Slika 4.6 Zahtev za rutom (RREQ) upit u AODV protokolu. S i D su izvorišni i odredišni čvorovi respektivno.



Slika 4.7 Route-odgovor (RREP) paket u AODV protokolu. S i D su izvorišni i odredišni čvorovi respektivno

Na svakom središnjem čvoru, put do odredišta je na raspolaganju ukoliko čvor ima "noviju" putanju od one koja je u RREP paketu. U slučaju da je RREQ poruka od srednjeg čvora, srednji čvor takođe šalje RREP do odredišta. Na ovaj način može se dozvoliti da putanja čvora ide u oba smera. U slučaju da je čvor dobio novu putanju (pomoću RREQ ili od strane RREP) i već je dobio informaciju za nju, tabela će biti ažurirana.

Opciono, *route_reply* potvrdna poruka (RREP-ACK) može biti poslata od začetnika RREQ koji priznaje prijem RREP. RREP-ACK poruka nema nikakve informacije za promenu. Pored ovih ruting poruka, *route_error* (RERR) poruka se koristi da obavesti ostale čvorove da neki čvorovi više nisu dostupni zbog prekinutih veza. Kad se reemituje RERR, u stvari se dodaju nedostupne destinacije na koje čvor ne može proslediti poruku. Dakle, promena informacija u RERR je spisak nedostižnih destinacija. Može se predvideti sledeće: u svakom skoku lista nedostupnih destinacija može ostati nepromenjena ili može postati podskup originalne. Pošto AODV nema bezbednosnih mehanizama, zlonamerni čvor M može da izvrši tipove napada koje se ne odnose samo na protokol:

- Lažno predstavljanje čvora S s falsifikovanim RREQ upitima s identičnom adresom validnih čvorova.

- Prilikom dostavljanja RREQ upita generisanih od čvora S da bi otkrio put do čvora D smanjuje *hop count* polje da bi povećao šansu da pronađe put između S i D čvora (ovo omogućava analizu saobraćaja između njih).
- Lažno predstavljanje čvora D s falsifikovanim RREP i svojom adresom kao odredišnom.
- Lažno predstavljanje čvora s falsifikovanim RREP koji tvrdi da je čvor destinaciona adresa.
- Selektivno uklanjanje određenih RREQ i RREP paketa. Ova vrsta napada je posebno teška za otkrivanje jer je efekat sličan greškama u prenosu.
- Falsifikovanje RERR potvrde koju čvor C šalje susedu D. RERR poruka ima vrlo visok *destination sequence number* (*dsn*) za jednu od nedostupnih destinacija, koja ima vrednost *U*. To može da izazove update *dsn* čvora D na *U* vrednost, što će sprečiti da čvor D dobije validan update.
- Prema specifikaciji AODV (Perkins et al., 1999), inicijator RREQ poruke može objaviti mnogo veći *dsn* od stvarne vrednosti. Ovo omogućava veoma jednostavan napad, gde je napadač u stanju da podesi broj sekvence čvora do željene vrednosti prilikom slanja samo dve RREQ poruke.

***Secure ad hoc on-demand distance vector routing protocol* (SAODV):** osnovni zadatak SAODV protokola jeste da se obezbedi AODV (Zapata et al., 2002). Ideja je da se koristi SAODV potpis za autentikaciju većinu RREQ i RREP upita i da koriste heš lance za potvrdu identiteta *hop-counta*. SAODV kreira ekstenziju potpisa za AODV. Mrežni čvorovi autentifikuju AODV ruting pakete sa SAODV potpisom, što sprečava izvesne napade lažnog predstavljanja. U SAODV, RREQ paket uključuje ekstenziju potpisa (RREQ-SSE). Inicijator bira maksimalnu vrednost *hop-counta*, prema očekivanom mrežnom dijametru i generiše jednosmerni heš lanac čija je dužina jednaka maksimalnom *hop-count* plus jedan. Ovaj jednosmerni lanac koristi se kao metrički authentikator, slično kao heš lanac kod SEAD protokola (Hu et al., 2002b). Ovu vrednost zovemo hop-count autentikator (HCA). Na primer, ako su vrednosti hash lancah_0, h_1,….., h_Ni ako su generisane tako da je$h_i = H[h_i+1]$, onda hop-count authentikator h_i odgovara broju skoka od $N - i$.

Prilikom prosleđivanja RREQ poruka u SAODV, čvor vrši autentikaciju RREQ paketa kako bi proverio validnost svakog polja. Zatim vrši suzbijanje duplikata da bi se prosledio samo jedan RREQ za svako otkrivanje putanje. Kada RREQ pronađe hop, proverava autentifikaciju pomoću RREQ-SSE. Ako je RREQ validan, hop vraća RREP kao u AODV. RREP-SSE (Route Replay Single Signature Extension) obezbeđuje autentifikaciju za RREP. Kao u RREQ, jedina promenljiva je polje hop-count. Kao rezultat toga, RREP je osiguran na isti način kao i RREQ.

Čvor dostavljanjem RREP-a proverava ekstenziju potpisa. Ako je potpis validan, onda dostavlja ruting tabele originalnom izvoru RREP.

$$S \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}0,h_N \rangle$$

$$A \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}1,h_{N-1} \rangle$$

$$B \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}2,h_{N-2} \rangle$$

$$C \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}3,h_{N-3} \rangle$$

$$D \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}0,h'_N \rangle$$

$$C \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}1,h'_{N-1} \rangle$$

$$B \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}2,h'_{N-2} \rangle$$

$$A \rightarrow * : \langle (RREQ,id,seq,D,oldseq_{D,}h_0,N)_{K^-_{s,}}3,h'_{N-1} \rangle$$

SAODV omogućava odgovore od srednjih čvorova kroz upotrebu *route reply double signature extension* (RREP-DSE). Srednji čvor koji odgovara RREQ uključuje RREP-DSE. Ideja je da se uspostavi put do odredišta, srednji čvor mora imati prethodno prosleđen RREP s odredišta. Ako srednji čvor ima uskladišten RREP i potpis, onda se može vratiti isti RREP ako je redni broj u tom RREP-u veći nego redni broj naveden u RREQ-u. Međutim, neka od polja tog RREP-a, posebno "the life-time" polje, više nisu važeća. Kao rezultat, drugi potpis, koji izračunava srednji čvor, koristiće se za potvrdu ovog polja.

Da bi dopustio odgovore zasnovane na informaciji iz RREQ paketa, inicijator uključuje odgovarajući potpis za RREP paket korišćenjem RREQ-DSE. Konceptualno, RREQ-DSE je RREQ i RREP smešteni u jednom paketu. Za smanjenje opštih troškova SAODV koristi zapažanje da se RREQ i RREP polja znatno preklapaju. Konkretno, RREQ-DSE treba da uključi parametar kao što su prefiks veličine i neka rezervisana polja, zajedno s važećim potpisom za RREP koristeći te vrednosti. Kad čvor prosledi RREQ-DSE, on kešira put i potpis na isti način kao da je prosledio RREP.

SAODV takođe koristi potpis da zaštiti *route_error* poruku(RERR) koja se koristi u održavanju putanje. U SAODV, svaki čvor potpisuje RERR koji emituje, bilo da potiče od RERR ili ga prosleđuje. Čvorovi koji sprovode SADOV ne menjaju njihovo odredište, redni broj informacije kad primaju RERR zbog toga što odredište ne

potvrđuje odredište rednog broja. Sledeća formula pokazuje primer SAODV-ove rute održavanja.

$$B \rightarrow A : (RERR,D,seq_D)_{K^-B}$$

$$A \rightarrow S : (RERR,D,seq_D)_{K^-A}$$

4.8.5. Secure routing protocol (SRP)

Siguran protokol rutiranja koji može biti primenjen na nekoliko postojećih protokola rutiranja (posebno na DSR (Johnson et al.,2007)) je predložio Paradimitratos (Papadimitratos et al., 2002). To je *on-demand* protokol rutiranja koji obuhvata osnovne karakteristike reaktivnog rutiranja. Paketi u SRP imaju produžena zaglavlja koja su pričvršćena na RREQ i RREP poruke. Protokol ne pokušava da obezbedi RERR pakete; umesto toga izvodi *route-maintenance* funkciju koja održava rutu bezbednom pomoću sigurnosnog protokola za prenos poruka. SRP koristi sekventni broj u RREQ-u i RREP-u da osigura svežinu, ali ovaj sekventni broj može jedino biti proveren na samom cilju. SRP zahteva bezbednost povezivanja jedino između komunikacijskih čvorova i korišćenja ovog bezbednosnog povezivanja za potvrđivanje RREQ-a i RREP-a kroz upotrebu *message autetnication codes* (MACs) kodova. Na cilju, SRP može otkriti bilo koje izmene RREQ-a, i kod izvornog čvora može otkriti izmene RREP-a.

U SRP *route requests* (RREQs), proizvedeni od izvornog čvora S, zaštićeni su MACs kodovima prebrojanih koristeći ključ podeljen s ciljem *T*. Zahtevi su emitovani svim *S* susedima. Svaki sused koji primi zahtev prvi put dodaje svoj identifikator zahtevu i reemituje ga. Srednji čvor takođe izvodi istu akciju. MAC u zahtevu nije proveren zato sto samo *S* i *T* znaju koji ključ je korišćen za izračunavanje. Kad zahtev dostigne cilj *T*, njegov MAC proverava *T*. Ako je važeći, onda se pretpostavlja da su svi susedni parovi čvorova na putu RREQ-a susedi. Takve putanje nazvane su važeće ili verodostojne rute. Meta *T* zamenjuje MAC od važećeg RREQ-a za MAC izračunat istim ključem koji potvrđuje rutu. Ovo je onda vraćeno nazad *S* koristeći obrnutu rutu. Na primer, RREQ koji dostiže srednji čvor $X_j$je u sledećoj formi:

$$msg_{S,T,rrep} = (rrep,S,T,id,sn,X1,X2,.....X_p,mac_S)$$

gde je *id* nasumično generisan identifikator rute, *sn* je broj sesije i mac_s, MAC je od (*rrep,S,T,id,sn)* izračunat pomoću *S* koristeći taster deljen sa *T*, X_1, ……., X_p, T je otkrivena ruta, nakon toga RREP za cilj *T* ima sledeću formu za sve središnje čvorove X_j, $1 \leq j \leq p$:

$$msg_{S,T,rrep} = (rrep,S,T,id,sn,X1,X2,.....X_p,mac_T)$$

gde je mac_TMAC izračunat s T s ključem deljenim sa S na polje za poruku koja ga prethodi. Srednji čvorovi trebalo bi da provere RREP zaglavlje (uključujući njegov *id* i *sn*) i da su oni susedi s dva od njihovih suseda na ruti pre slanja RREP uzvodno.

SRP ne pokušava da spreči neovlašćenu izmenu polja koja su obično izmenjena tokom prosleđivanja ovih paketa. Na primer, čvor može slobodno ukloniti ili izmeniti listu čvorova u RREQ paketu koji prosleđuje. Pošto SRP zahteva sigurnosnu vezu između čvorova koji komuniciraju, on koristi izuzetno lagane mehanizme da bi sprečio druge napade. Na primer, da bi ograničili poplavljivanje, čvorovi beleže stopu na kojoj svaki sused prosleđuje RREQ pakete i daje prioritet zahtevu paketa poslatom kroz suseda koji ređe prosleđuje zahtev paketima. Takvi mehanizmi mogu osigurati protokol kad je prisutno nekoliko napadača. Međutim, takve tehnike obezbeđuju sekundarne napade kao što je slanje lažnih RREQ paketa da bi se smanjila efikasnost čvora autentičnog RREQ-a. Dodatno, takve tehnike stvaraju problem pohlepnih čvorova. Na primer, čvor koji ne prosleđuje RREQ pakete obično postiže bolje rezultate zato što je generalno manje opterećen, i nema potrebe da koristi dodatnu energiju da prosledi pakete od drugih čvorova. U SRP, pohlepan čvor zadržava ove prednosti i, dodatno, dobija veći prioritet kad on inicira otkrivanje rute.

4.8.6. ARIADNE

Ariadne, siguran on-demand protokol rutiranja za ad hok mreže (Hu et al., 2002a) je siguran *on-demand* protokol rutiranja zasnovan na DSR (Dynamic Source Routing) protokolu (Johnson et al., 2007). Protokol može da izdrži kompromis čvora i oslanja se samo na vrlo efikasan simetrični kriptografski ključ. Ariadne može potvrditi ruting poruku koristeći jednu od tri šeme: (I) podeljena tajna između svakog para čvorova, (II) podeljene tajne između komunikacijskih čvorova kombinovanih s prenosom potvrđivanja koristeći TESLA (Perrig et al., 2001) i (III) digitalni potpisi. U ovom delu se opisuje Ariadne s TESLA, efikasnu šemu potvrde verodostojnosti koja zahteva slobodno vreme sinhronizacije. Koristeći *pair-wise* deljenih ključeva, protokol izbegava potrebu za vremenskom sinhronizacijom. Ariadne otkriva putanje reaktivno (on-demand). Kroz rutu otkrivanja koristi i da otkrije rutu podataka paketa ka njihovom odredištu. Svaki prosleđen čvor pomaže vršeći održavanje rute da otkrije probleme sa svakom izabranom rutom.

Otkrivanje rute (putanje): Plan protokola objašnjen je u dve faze: (I) mehanizam je predstavljen tako da omogućava meti čvora da proveri autentičnost RREQ-a i (II) i efikasna *per-hop* heš tehnika, opisana kako proverava da li neki čvor nedostaje s liste čvorova u RREQ-u. Pretpostavljamo da pokretač čvora S obavlja otkrivanje putanje za metu čvora D i da oni dele skrivene ključeve K_{SD}i K_{DS}, respektivno za poruku potvrđivanja u svakom smeru.

- *Target authenticates route request:* Da bi ubedio metu o legitimnosti svakog polja u RREQ-u, inicijator jednostavno uključuje MAC kod dobijen s ključem K_{SD} preko jedinstvenih podataka – na primer, "*timestamp*". Meta može lako potvrditi putanju tražene autentičnosti i svežine koristeći deljeni ključ K_{SD}. U otkrivanju putanje, inicijator želi da potvrdi svaki pojedinačni čvor u listi čvorova RREP-a. Drugi uslov jeste da meta može potvrditi svaki čvor u listi RREQ-a, tako da će vratiti RREP samo duž putanje koja sadrži legitimne čvorove. Svaki skok potvrđuje novu informaciju u RREQ koristeći svoj trenutni TESLA ključ. Meta čvora odbija RREP dokle srednji čvorovi mogu da otpuste odgovarajuće TESLA ključeve. TESLA bezbednosti uslov se proverava na ciljnom čvoru, a meta uključuje MAC i RREP da potvrdi da je stanje bezbednosti ispunjeno.
- *Per-hop hashing:* Potvrđivanje identiteta podataka u rutiranju poruka nije dovoljno zato što napadač može ukloniti čvor iz čvorne liste u RREQ-u. Jednosmerne haš funkcije koriste se da potvrde da nijedan skok nije izostavljen – pristup koji je nazvan per-hop hashing. Da bi promenio ili uklonio prethodni skok, napadač mora ili da čuje RREQ bez tog navedenog čvora, ili mora da bude u stanju da izokrene jednosmernu hash funkciju. Zbog efikasnosti, autentikator može biti uključen u hash vrednosti koje su bile u RREQ-u.

Održavanje putanje: Održavanje putanje u Ariadne je zasnovana na DSR protokolu. Čvor koji prosleđuje paket na sledeći nivo duž izvornog puta vraća RERR paket originalnom pošaljiocu ako nije u mogućnosti da dostavi paket na sledeći nivo posle ograničenog broja reemitovanih pokušaja. Da bi sprečio neovlašćene čvorove od slanja RERR-a, mehanizam treba da bude na mestu u kome pošiljalac treba da potvrdi RERR poruke. Svaki čvor na putu povratka do izvornog čvora prosleđuje RERR poruku. Ako je potvrđivanje kasni ili je odloženo – na primer, kad se koristi TESLA – svaki čvor koji će biti u stanju da potvrdi RERR poruku memoriše je sve do autentikacije.

Avoiding routing misbehaviour: goreopisan Ariadne protokol podložan je napadaču koji se zadesi duž otkrivene putanje. Posebno, tamo bi trebalo da se nalazi mehanizam koji je u stanju da odredi da li srednji čvorovi prosleđuju pakete koje su oni zahtevali. Da bi izbegli dalje korišćenje zlonamernih ruta, rute se biraju na osnovu njihovog prethodnog rada u paketu prosleđivanja. Šema se oslanja na povratne informacije o tome koji paketi su uspešno isporučeni. Povratna informacija može biti primljena kroz dodatni *end-to-end* mrežni sloj poruke ili iskorišćavanjem svojstava transportnih slojeva kao što su TCP s odabranim priznanjima (Mathis et al., 1996). Ovaj povratni pristup je sličan onom koji je korišćen u IPv6 za detekciju nedostižnih suseda (Narten et al., 2007). Čvor s više ruta ka jednom odredištu može izdvojiti deo paketa koji je potekao da bi bio poslat duž svake rute. Kad se bitno manji deo paketa, koji je poslat

duž svake rute, uspešno isporuči, čvor može početi da šalje manji deo svojih ukupnih paketa ka tom odredištu duž te rute.

4.8.7. Security Enhanced AODV (SEAODV)

Security Enhanced AODV (SEAODV) protokol rutiranja predložen je u Li et al., 2011. SEAODV upotrebljava Blumov ključ *pre-distribution* šeme (Blom, 1985) da bi izračunao *pair-wise transient key (PTK)* od poboljšane *hello* poruke i zatim koristi utvrđen PTK da podeli *group transient key* (GTK). PTK i GTK se koriste za potvrđivanje emisije ruting poruke. U WMNs-u, jedinstven PTK se deli od svakog para čvorova, dok se GTK deli tajno između čvora i svih njegovih *one-hop* suseda. MAC kod je priložen kao proširenje originalnom AODV protokolu da bi garantovala autentičnost i integritet poruke u *hop-by-hop* modu. SEAODV koristi Blumov ključ *pre-distribution* šeme.

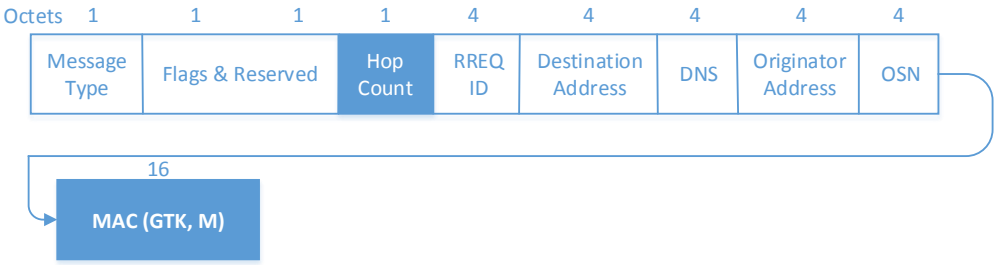
Blumov ključ predistribucione šeme: Blumov ključ predistibucije se primenjuje u sprovođenju procesa razmene ključa. Blumov *t* siguran ključ predistribucione šeme je sledeći (Blom, 1985; Du et al., 2003). Blumova predistribuciona šema se zasniva na *(N, t + 1)maximum distance separable* (MDS) linearnih kodova. U ovoj šemi, pre nego što je mreža raspoređena, centralni autoritet prvi konstruiše *(t + 1) x N* javnu matricu *P* nad konačnim poljem $GF_{(q)}$, gde je *N* veličina mreže. Onda centralni autoritet bira nasumice *(t + 1) x (t + 1)* simetričnu matricu *S* preko $GF_{(q)}$, gde je *S* skrivena i jedino poznata centralnom autoritetu. *N x (t + 1)* matrica $A = (S \,.\, P)^T$ je izračunata, gde (.) označava premeštenog operatera. Centralni autoritet opterećuje *i*-ti red i *i*-tu kolonu *P* do čvora *i*, za *i=1,2….n*. Kad čvor *i* i *j* treba da uspostave deljeni ključ, oni prvorazmene njihove kolone *P*, i onda čvor *i* računa ključ K_{ij} kao proizvod sopstvenog reda *A* i *j*-te kolone *P*, i čvor *j* računa K_{ij} kao proizvod sopstvenog reda *A* i *i*-te kolone *P*. Pošto je *S* simetrično, lako je videti da:

$$K=A\cdot P=(S\cdot P)^T\cdot P=^T\cdot S^T\cdot P=P^T\cdot S\cdot P=(A\cdot P)^T=K^T$$

Par čvorova *(i,j)* koristi $K_{ij} = K_{ji}$ kao deljeni ključ. Blumova šema ima *t*-sigurnu osobinu. Podrazumeva se da u mreži od *N* čvorova uređaji u sprezi manjoj od *t + 1* čvorova ne mogu otkriti bilo koji ključ deljen s drugim parom čvorova. Ovo je zato što su najmanje *t* redova od *A* i *t* kolona od *P* moraju da reše tajnu simetrične matrice *S*. Memorija troškova po čvoru u Blumovoj šemi je *t + 1*. Da bi se garantovala savršena sigurnost u WMN s *N* čvorova, treba da se koristi *(N - 2)* – sigurna Blumova šema, što znači da je memorija troškova po čvoru *N – 1*. Stoga Blumova šema može obezbediti jaku bezbednost u mrežama malih dimenzija.

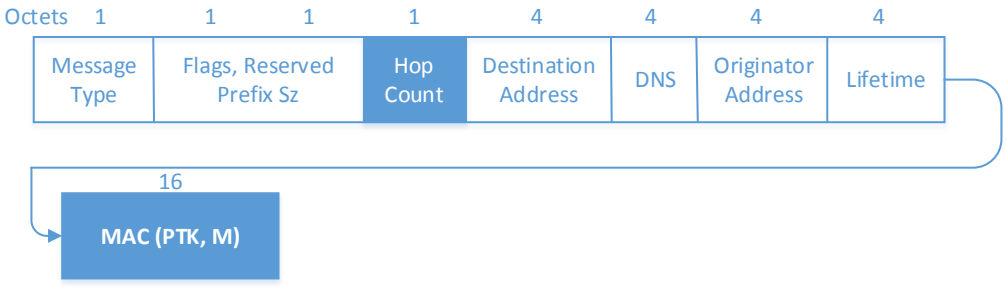
SEAODV protokol: SEAODV je izgrađen na AODV protokolu. To zahteva da svaki čvor u mreži održava dva ključa hijerarhije. Jedan je emitovana hijerarhija ključa koja obuhvata sve emitovane ključeve od svojih aktivnih *one hop* suseda. Drugi je nazvan *unicast* hijerarhija, koja skladišti sve tajne *pair-wise* ključeve koje ovaj čvor deli sa svojim *one hop* susedima. Svaki čvor koristi ključeve u svom emitovanju ruting poruka (RREQ poruke) od njegovih *one hop* suseda i primenjuje tajnu *pair-wise* ključeva u unikast hijerarhiji da bi proverio dolazne poruke kao što su RREP poruke. Različite karakteristike protokola:.

- **Napredne (enhanced) hello poruke:** u AODV, hello poruka se emituje od svakog čvora u svom *one hop* susedstvu. U SEAODV, dve *ehnanced hello* poruke su definisane sledeću ideju predstavljenu u Jing et al., 2004. Svaki čvor ugrađuje svoju kolonu javne matrice P u svoju *ehnanced hello* RREQ poruku. Da bi garantovali dvosmernost veza, susedni čvorovi koji primaju *hello* RREQ odgovaraju *ehnanced hello* RREP.
- **Razmena javne Seed_P i GTK koristeći ehnanced hello poruku:** tokom ključne predistribucione faze, svaki legitiman čvor u WMN zna i skladišti javni *Seed_P* (seed kolone javne matrice P) i odgovarajući privatni red generisane matrice A. Ceo proces razmene je opisan u tri koraka: (a) razmena *Seed_P* javne matrice P, (b) izvođenje PTK i (c) razmena GTK. U razmeni Seed_P faze, svaki čvor traži svoj javni Seed_P od svog osnovnog ključa i emituje *enhanced hello* RREQ poruku. Na završetku ovog koraka, svaki čvor u mreži poseduje javni Seed_P od svih svojih *one-hop* suseda. U izvođenju PTK faze, svaki čvor koristi Seed_P primljen od suseda i čvor odgovara privatnom redu matrice A da bi izračunao PTK. Na završetku ovog koraka, svaki čvor je sačuvao javni Seed_P svojih suseda i izveo PTK koji deli sa svakim od svojih *one-hop* suseda. U razmeni GTK faze, po dobijanju *hello* RREQ od čvora X, čvor Y (čvor X je sused) šifrira GTK_Y sa svojim privatnim PTK_Y i unicasted odgovarajuću *hello* RREP poruku nazad do X. Šifrovan GTK_Y je takođe uključen u sastav *hello* RREP poruke. Kada X primi *hello* RREP od Y, X primenjuje svoje privatne PTK_X da dešifruje GTK_Y i skladišti ga u bazu podataka. Isti proces primenjuje se na čvoru Y takođe. Konačno, svaki čvor poseduje GTK ključeve od svih svojih *one hop* suseda i grupu skrivenih *pair-wise* PTK ključeva koje deli sa svakim od svojih *one-hop* suseda.



Slika 4.8 Struktura RREQ poruke u SEAODV protokolu

- **Obezbeđivanje otkrivanja putanje:** da bi se osigurala *hop-by-hop* autentifikacija, svaki čvor mora da proveri dolaznu poruku od svojih *one-hop* suseda pre reemitovanja poruka. Odnos poverenja između svakog para čvorova oslanja se na deljeni GTK i PTK. Proces otkrivanja putanje SEAODV je sličan AODV protokolu, izuzev za MAC produžetak priložen AODV poruci. Struktura RREQ-a u SEAODV predstavljena je na slici 4.8. MAC je kompjuterizovan za poruku M koristeći GTK čvora koji treba da emituje RREQ do svojih *one-hop* suseda. Kad čvor želi da otkrije putanju do određene destinacije, on emituje modifikovanu RREQ poruku svojim susedima. Prijemni čvor izračunava odgovarajuće MAC vrednosti primljene poruke ako čvor poseduje GTK pošiljaoca. Prijemni čvor onda ažurira promenljivo polje i njegovu ruting tabelu i zatim postavlja obrnut put nazad do izvora, snimanjem suseda od koga je primio RREQ. Konačno, čvor izračunava MAC od ažuriranih RREQ sa svojim GTK i pridaje MAC vrednost kraju RREQ pre nego što je poruka reemitovana njihovim susedima.
- **Obezbeđivanje putanje podešavanja:** odredišni čvor ili srednji čvor generiše modifikovani RREP i emituje ga nazad na sledeći hop od koga je dobili RREQ. Pošto je RREP poruka potvrđena na svakom skoku koristeći PTK, protivnik nema priliku da preusmeri promet (Slika 4.9).

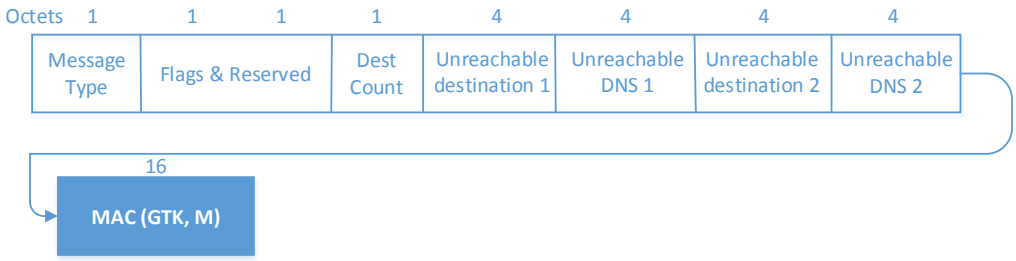


Slika 4.9 Struktura RREP poruke u SEAODV protokolu

Po prijemu RREP-a od čvora *Y*, čvor *X* proverava da li je *PTK_YX* u svojoj grupi *PTK*. Ako jeste, onda čvor *X* izračunava *MAC'(PTK_XY, M)* i upoređuje s *MAC(PTK_YX, M)*

koji je primljen od čvora *Y*. Ako *MAC'(PTK_XY, M)* odgovara *MAC(PTK_YX, M)*, primljeni RREP se smatra autentičnim. Čvor X onda ažurira *hop-count* polje u RREP-u i svoju tabelu putanje postavlja prosleđivanje puta ka odredištu. Čvor *X* takođe traži odgovarajući PTK koji deli sa svojim sledećim skokom na koji će nov RREP biti prosleđen izvoru. Čvor *X* onda koristi PTK da izgradi novi MAC i dodaje se novoj RREP poruci. Inače, primljeni RREP se smatra neautentičnim.

- **Obezbeđivanje otkrivanja rute:** čvor generiše RERR poruku ako prima podatke paketa predodređene drugom čvoru za koji nema aktivnu rutu u svojoj tabeli, ili čvor otkriva pokvaren link za sledeći hop aktivne rute ili čvor prima RERR poruku od suseda za jednu ili više aktivnih ruta. Struktura modifikovane RERR poruke predstavljena je na slici 4.10. MAC polje u modifikovanoj RERR poruci se izračunava primenom čvornog *GTK* na ceo RERR paket. Po prijemu emisije RERR poruke iz čvora Y, čvor X prvo proverava da li ima *GTK_Y*. Ako ima, čvor *X* onda izračunava *MAC'(GTK_Y, M')* i upoređuje ga s primljenim MAC-om. Ako se dva MAC-a podudaraju, čvor X traži svoju ruting tabelu i pokušava da identifikuje pogođene rute (nova grupa nedostupnih destinacija) koje koriste čvor Y kao svoj sledeći hop zasnovan na nedostupnoj listi destinacije primljene od Y. Ako nema rute koja je pogođena u čvoru *X's* tabele, *X* jednostavno ispušta RERR poruku i počinje ponovo da sluša kanal. Čvor X takođe odbacuje RERR poruku ako uspe da pronađe GTK_Y, ili *MAC'(GTK_Y, M')* ne odgovara onom primljenom od čvora *Y*.



Slika 4.10 Struktura RERR poruke u SEAODV protokolu

SEAODV je podložan napadu RREQ. Ipak, pošto potvrđuje RREQ od čvorova koji se nalaze u listi aktivnih *one-hop* suseda, otkrivanje napada će biti brzo. Pošto se GTK i PTK koriste da osiguraju emisiju poruka, integritet poruka je zaštićen od MAC. Protokol je otporan na RREP petlju usmeravanja napad i preusmeravanje rute. RERR fabrikovan napad ima minimalan uticaj na SEAODV protokol pošto prijemni čvor potvrđuje autentičnost RERR poruka koje stižu samo od njihovih *one-hop* suseda. Pošto zlonameran čvor može samo da prosleđuje ponavljanje RERR poruke koje stižu od *one-hop* suseda prijemnih čvorova, lansiranje RERR izmišljenog napada postaje naročito teško.

Sledeća tabela prikazuje karakteristike WMN, odnosno MANET protokola.

Protocols	Secret Keys	MAC	Digital Signature	Hash Chain	Cryptographic mechanism	Assumptions	Verification mechanism
Ariadne	Secret MAC keys K_{SD} between sender and receiver	MAC K_{SD}	—	TESLA. keys authenticate	—	Nodes have loosely synchronized clocks	MAC verification mechanism
SAOVD	Public and private key pair for each node		Sender uses digital signature to sign the messages	One way hash chain to authenticate hop		Network should he Key Distribution System	Digital signature verification mechanism
SEAD	Initial secret key $K_N$for hash function	—	—	Authenticates the sequence number and routing	—	Secure way of delivering initial secret key K_N	Hash chain verification
SDSDV	Different Pair-wise K_{ij} shared secret key between all the nodes	Node i sends MAC K_{ij} to node j	—			Public Key Infrastructure	MAC verification mechanism
SLSP	Public and private key pair for each node	MAC	—		Threshold cryptography	Single network interface per node	Threshold cryptography for key certification; MAC verification
SRP	SA between source and destination	MAC calculation with K_{ST}	—		—	Secure way of delivering the SA	MAC verification mechanism
Secure protocol resilient to Bizantaine failures	Pair-wise secret key established on demand	—	Digital signature is used to authenticate the source			Public key infrastructure or CA	Digital signature verification mechanism
ARAN	Public and private key pair for each node				Public key cryptography	Trusted certificate server	Public key cryptography verification mechanism
SPAAR	Public and private key pair for each node; Group neighborhood key	—			Public key cryptography	Trusted certificate server	Public key cryptography verification mechanism

Tabela 4.2 Komparacija bezbednosnih protokola rutiranja u WMN mrežama

4.9. IEEE 802.11s bezbednosni standard

IEEE 802.11s je dopuna za IEEE 802.11 mesh mreže koja definiše kako se bežični uređaji međusobno povezuju da bi kreirali WLAN mesh na osnovu statičke topologije i ad hok mreža. IEEE 802.11 je skup standarda koji regulišu metode prenosa bežičnog umrežavanja. Najčešće verzije standarda koje se danas koriste su 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac. 802.11s proširuje IEEE 802.11 MAC standard definisanjem arhitekture i protokola koji podržavaju emitovanje multikast i unikast dostavu korišćenjem "radio aware metrics over self-configuring multi-hop topologies".

802.11s inherentno zavisi od standarda (802.11a, 802.11b, 802.11g, 802.11n, 802.11ac) kao nosioca stvarnog saobraćaja. Neki protokoli rutiranja raspoloživi za fizičku mrežnu topologiju za 802.11s standard podrazumevaju Hybrid Wireless Mesh Protocol (HWMP). Međutim, druge mesh ili ad hok mreže podržavaju dinamičko link-state (OLSR, B.A.T.M.A.N) ili statičko (OSPF, WDS) rutiranje.

4.9.1. 802.11 mesh arhitektura

802.11s mrežni uređaj se označava kao Mesh Station (mesh STA). Mesh STA formira mesh linkove s drugim uređajima, preko putanja koje se utvrđuju pomoću protokola za rutiranje. Mesh STA su pojedinačni uređaji koji koriste mesh servise za komunikaciju s drugim uređajima u mreži.

4.9.2. 802.11s protokoli

802.11s se definiše na osnovu HWMP protokola rutiranja, ali omogućava proizvođačima da koriste i alternativne protokole. HWMP je inspirisan kombinacijom AODV i tree-based rutinga. 802.11s uključuje mehanizme za deterministički pristup mreži, okvir za kontrolu zagušenja i uštedu energije.

4.9.3. 802.11s bezbednost

U mesh mreži ne postoji definisana uloga, ne postoje klijenti, server, nema inicijatora i respondira. Bezbednosni protokol koji se koriste u mesh mrežama moraju da budu istinski peer-to-peer, gde svaka strana može inicirati drugu ili se obe strane mogu pokrenuti u isto vreme.

U mesh mreži 802.11s definiše sigurnosne password-based provere autentičnosti i proveru uspostavljanjem ključa koja se naziva Simultaneous Authentication of Equals (SAE). SAE se zasniva na Diffie–Hellman razmenu ključeva korišćenjem konačne ciklične grupe koje mogu da budu primarne ili da se zasnivaju na eliptičnoj krivi, tako da je sposoban da pokrene obe stanice istovremeno. Zbog pairwise (uparene) enkripcije,

802.11s ne obezbeđuje end-to-end enkripciju. Takođe, problem kod korišćenja Diffie–Hellman razmene ključeva jeste da ne postoji mehanizam potvrde identiteta.

4.10. Izazovi u WMN mrežama

Postoji mnoštvo izazova u odnosu na ad hok umrežavanje, a naročito kod mobilnog ad hok umrežavanja. Prvo, postoje usmeravajući izazovi koji se tiču preklapanja i petlji, drugo, postoji održavanje energije jer ad hok petlje obično rade na baterije i treće, postoji pitanje bezbednosti (Sanzgiri et al., 2002).

4.10.1. Usmeravajuće preklapanje

Usmeravajuće preklapanje je termin koji opisuje nagle promene putanje kad postoji više mogućih ruta između dva čvora. Ponekad dve rute mogu biti gotovo podjednako dobre i mogu tako postojati neko vreme, što može dovesti do toga da se rute u petljinoj tabeli konstantno menjaju između njih dva. U takvim scenarijima usmeravajući protokol može da reši problem tako što neće promeniti rute, osim ako druga ruta nije bitno bolja ruta ili ako je bila zamalo bolja duži vremenski period. Koje rešenje je bolje je teško odlučiti i jedno od mnogih pitanja – zašto postoji ogromna količina konkurentnih ad hok usmeravajućih protokola – ostaje nerešeno.

4.10.2. Usmeravajuće petlje

Usmeravajuća petlja postoji ako put između dva čvora prolazi kroz jedan ili više čvorova na trasi više nego jednom. Kad čvorovi imaju pogrešnu, nepotpunu ili samo drugačiju usmeravajuću informaciju jedan od drugog, onda dva čvora u mreži mogu izabrati da usmeravaju iste pakete kroz različite staze, što može dovesti do toga da se ruta vremenom vrati do čvorova koji su već primili i forvardovali pomenute pakete.

U MANET-u, gde se putevi često mogu menjati, šanse da dva čvora imaju različite usmeravajuće informacije su velike. Usmeravajuće petlje stoga mogu postati ozbiljan problem u MANET-ima, što sugeriše da bi implementacije trebalo da koriste protokol koji ima neke mehanizme da bi se minimizovale

Šanse usmeravajućih petlji, i/ili detektovanje usmeravajućih petlji i njihovo uklanjanje posle toga.

4.10.3. Konzervacija energije

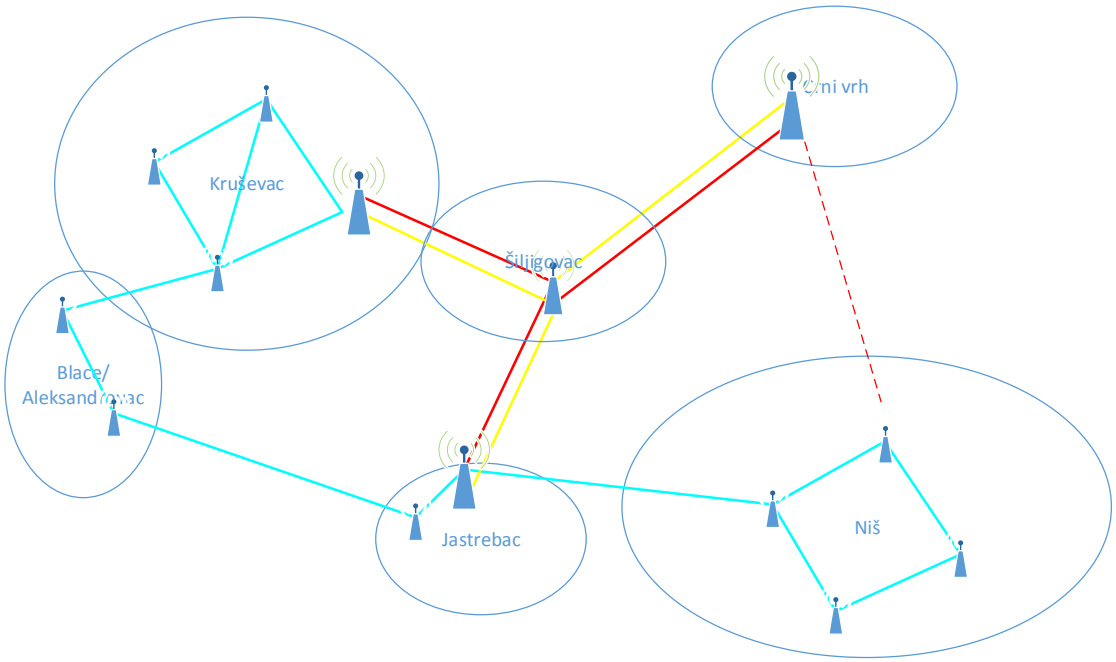
Mobilni čvorovi u MANET potrošnji energije mogu postati velika prepreka. Pošto proaktivni usmeravajući protokoli redovno šalju usmeravajuće informacije, čak i kad nisu potrebni, oni mogu trošiti energiju u odnosu na relativne protokole koji

obračunavaju samo usmeravajuće putanje kad se aplikacija s podacima šalje kroz mrežu.

Na ovo pitanje će veoma uticati i sprovođenje bezbednosti na vrhu ad hok usmeravajućeg protokola. Ukoliko, na primer, svaki paket treba da bude digitalno potpisan, ključ za enkripciju operacija će se vršiti na svakom paketu – uz upotrebu mnogo energije. Kad se projektuje rešenje za potvrdu identiteta, treba obratiti pažnju na ovo pitanje.

4.11. Arhitektura i algoritmi za multi-kanal WMN mreže (Load balancing)

U bežičnim mrežama postoji nekoliko kanala koji se ne preklapaju u 5 GHz i2.4 GHz spektru, većina IEEE 802.11 baziranih multihop ad hok mreža danas koristi jedan kanal (Raniwala et al., 2005). Kao rezultat toga, ovakve mreže retko mogu u potpunosti da iskoriste ukupni raspoloživi propusni opseg u spektru na osnovu standarda. Predlog je da se svaki čvor WMN mreže ima više 802.11 mrežnih interfejs kartica a suština ovakve multihop WMN arhitekture je dodela kanala i ruting (slika 4.11). Dokazano je da svaki čvor sa dva bežična mrežna interfejsa može da poboljša protok za faktor jedan u poređenju s jednokanalnom ad hok arhitekturom.



Slika 4.11 Load balancing u IEEE 802.11 mreži

U ovakvim slučajevima, gejtvej čvor koordinira pokretanje ostalih čvorova u stablu da bi ostvario load balansing. Metrike koje su predložene za merenje opterećenosti linka su broj paketa ili broj putanja koje prolaze kroz određeni čvor u zoni njegove interferencije. Da bismo pronašli load-balancing putanju, merimo saobraćaj kroz svaki

čvor, izračunavamo slobodan propusni opseg na različitim linkovima kao i na gejtvej čvoru. Različiti linkovi u susedstvu mogu da rade na različitim kanalima, izračunava se opterećenje svakog kanala na osnovu čega se izračunava preostali raspoloživi protok na svakom linku. Za ovakav tip algoritma karakteristične su dve strategije – prosleđivanje svakog paketa na najmanje opterećeni čvor i prosleđivanje svakog paketa na najudaljeniji čvor, gde se dodatno uzima u obzir opterećenje ometanih čvorova i raspoloživi propusni opseg koji ima na raspolaganju svaki čvor.

I pored tehničkih dostignuća na fizičkom sloju, ograničeni propusni opseg ostaje urgentno pitanje za bežične mreže, naročito ako se uporedi s kablovskim mrežama. Ozbiljniji problem postoji kod multihop bežičnih mreža (WMN) zbog interferencije između uzastopnih skokova na istom putu kao i između suseda. Kao rezultat toga konvencionalne jednokanalne WMN mreže ne mogu adekvatno ispuniti zahtev za optimalnim protokom u last-mile bežičnim brodbend mrežama, a još manje zahteve kampus Eternet mreža. Pomoću čvorova s više mrežnih interfejsa, iskorišćavanjem nepreklapajućih kanala kojim dobijamo maksimalni propusni opseg u spektru, s odgovarajućim algoritmom za multikanal WMN mreže, efikasno rešavamo problem iskorišćavanja maksimalnog IEEE 802.11 propusnog opsega na fizičkom sloju.

4.12. Fault Tolerant algoritmi u WMN mrežama

WMN mreže ostvarile su značajan razvoj zbog brzog razmeštanja, lakog održavanja, niske investicije, bržeg nadograđivanja u poređenju s tradicionalnim bežičnim mrežama. Upotreba WMN mreža podstiče stvaranju ruting algoritama za prenos podataka na različitim osnovama. Tako imamo table-driven, on-demand ili hibridne algoritme. S obzirom na to da su WMN obično kičma mreže, za njih je karakteristično da bežična komunikacija nije konstantna. Zbog toga je važna uspešna veza prema linku ili čvoru. Ideja je da se za toleranciju greške koriste inovativni algoritmi uz pomoć antenskih nizova u samoj kičmi mreže (Gupta et al., 2011). Algoritam prvo detektuje čvor prema kome je izgubljena veza i na osnovu toga pokušava da uspostavi alternativnu vezu na osnovu antenskih nizova.

WMN mreže su infrastrukturne mreže s multihop bežičnom kičmom. One se sastoje od mreže rutera i mreže klijenata. Ruteri su retko mobilni, tako da formiraju kičmu WMN. Oni služe za pristup mesh i konvencionalnih klijenata. Nastale su zbog ograničenja ad hok mreža, jer se povećava propusni opseg i performanse. Ideja WMN mreža nije samo za pružanje usluga kao što su glas, video, Internet itd. Pristup na raznim lokacijama i unifikacija različitih mrežnih rešenja je jasan cilj prema kome se WMN mreže razvijaju.

Struktura WMN se sastoji od baznih stanica, kičme, linkova i mobilnih stanica. Problem je upravljanjem pokrivenosti između čvorova, bilo da su bazne stanice ili mobilni

čvorovi. Potrebno je da sistem postigne stabilnost i toleranciju na greške. Postoje dva glavna koraka u izradi sistema za toleranciju greške. Prvi je otkrivanje prisustva greške na različite načine. To je važan aspekt jer je za toleranciju greške bitan faktor njeno otkrivanje. Drugi je planiranje pokrivenosti u dometu smetnji.

Kičma WMN mreže može se sastojati od kolekcije radio-tehnologija. Greške se mogu desiti u samoj kičmi mreže ili u klijentskim vezama. To znači da moramo uspostaviti kontrolu greške između, na primer, klijentskih veza i baznih stanica istovremeno. Kičmu WMN mreže formiraju zajedničke konekcije baznih stanica. Konekcija između baznih stanica formira mesh koji upravlja čvorovima koji prelaze s jedne bazne stanice na drugu. Bazne stanice moraju održavati vezu između sebe, kao i klijentskih čvorova. Mora postojati radio-pokrivenost između čvorova i baznih stanica da bi se izgradila mreža za pružanje komunikacije.

Kičma konvencionalnih mreža je obično povezana žičanom infrastrukturom (optički, ISDN, Ethernet, HDSL i dr.), gde se lako uspostavlja kontrola, ali u WMN mrežama je veza bežična, pa moramo voditi računa o smetnji na baznoj stanici i pokrivenosti.

U WMN mrežama postoje dve kategorije grešaka. Prva je gubitak veze, gde čvorovi gube domet, a druga kad se gubi link između mesh tačaka. Gubitak linka možemo podeliti u dve kategorije – *single-link failure* i *multiple link failure*. Postoji tri pristupa za oporavak veze – šema zaštite, šema restauracije i hibridna šema. U šemi zaštite, dva (ili više) prekinuta puta biraju izvorni i odredišni čvor. Izvorni čvor prosleđuje podatke prema svim izabranim putanjama. Ako je veza prekinuta na bilo kojoj tački, odredišni čvor može dobiti podatke s bilo koje druge putanje. U restauracionoj šemi, prilikom detekcije greške, glavna putanja menja se rezervnom putanjom dinamički, gde se uključuje i neznatno kašnjenje u procesu oporavka. Hibridne šeme pribegavaju restauraciji ako zaštita ne uspe. Gubitak linka može prouzrokovati gubitak veze između različitih baznih stanica ili čvorova.

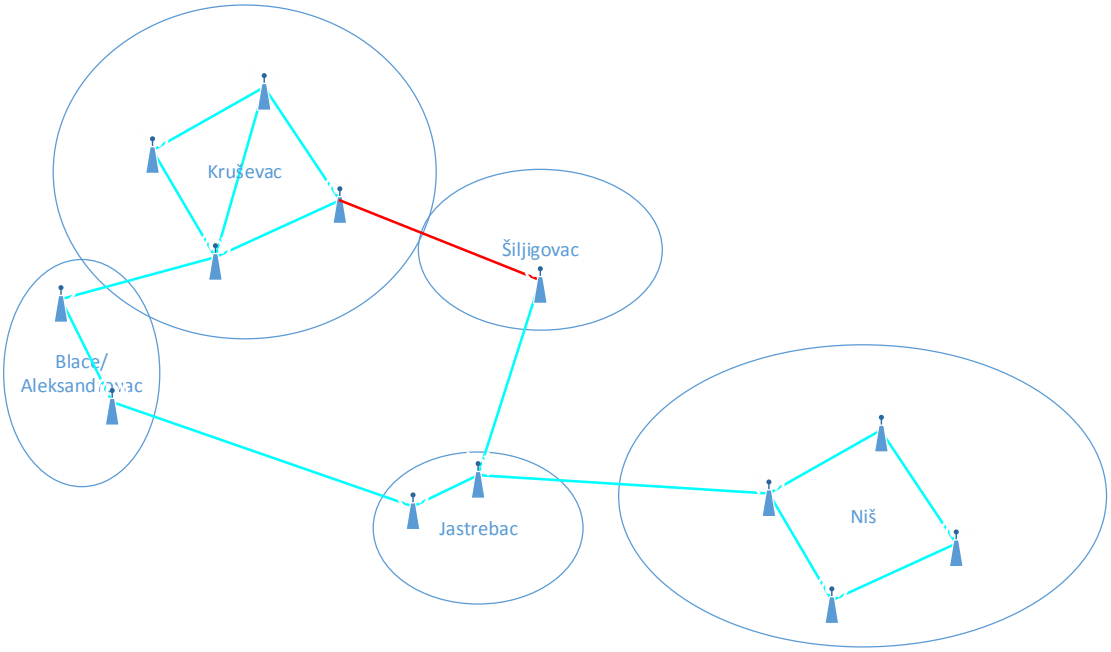
Za poboljšanje tolerancije greške pokrivenost kičme i *last mile* pokrivenost su značajne oblasti.

Za ovo su nam potrebne sledeće informacije:

1. Definisati pokrivenost servisima, tj. uslugama. To označava oblast koja povezuje mrežu između različitih čvorova. Ona nam pomaže da utvrdimo rešenja za povezivanje.
2. Planiranje i izbor lokacija za bazne stanice, koja je odgovarajuća za *setup*.
3. Održavanje informacija o pokrivenosti:
    a. Informacije o kičmi mreže
    b. Last mile informacije

Nakon utvrđivanja ovih informacija dobija se procena o radio-pokrivenosti i toleranciji greške.

Upotreba niza antena na baznim stanicama povećava efikasnost i toleranciju na greške u bežičnim sistemima (Slika 4.12).



Slika 4.12 Lokacija baznih stanica mesh mreže ITSNet provajdera

Kao primer imamo mesh mrežu kompanije ITSolutions.NET koja se sastoji iz mesh baznih stanica na sledećim lokacijama: Niš, Jastrebac, Kruševac, Aleksandrovac, Veliki Šiljegovac. Kičma mreže je prikazana na slici 4.12. U ovom slučaju kičma mreže je tolerantnija na greške, osim u slučaju ako imamo prekid između Niša i Jastrepca. Ali, dodavanjem nove bazne stanice s nizom antena na novoj lokaciji, s odgovarajućim algoritmom mreža bi postala tolerantnija na greške (Slika 4.12).

1. Pronaći stepen svakog čvora
2. Identifikovati čvorove koji imaju manji stepen od 3
3. Za svaki čvor na grafu:
a. Ukloniti čvor
b. Čekirati konekciju
c. Ako se grafik diskonektuje, onda

i. Postaviti novu baznu stanicu da bi se uspostavila konekcija
d. Kraj uslova
4. Kraj petlje

5. PERFORMANSE WMN PROTOKOLA RUTIRANJA

5.1. Pregled B.A.T.M.A.N. protokola

B.A.T.M.A.N. (A. Neumann et al., 2008) je sve popularniji usmeravajući protokol za bežične ad hok mreže, što pokazuje činjenica da je preuzet od Linux kernel net tree. Ime je skraćenica za *Bolji pristup mobilnom ad hok umrežavanju*. B.A.T.M.A.N.-a je nastao kao zamena Optimizovanog Link State protokola rutiranja (OLSR) zbog inherentnih teškoća koje taj protokol ima, kako je objašnjeno u nastavku.

5.1.1. Opis B.A.T.M.A.N. protokola

OLSR je proaktivni usmeravajući protokol, što znači da čvorovi koji tu učestvuju redovno međusobno razmenjuju usmeravajuće informacije. Prema B.A.T.M.A.N. programerima, problem sa OLSR jeste što svaki čvor u mreži izračunava čitav put usmeravanja, što je veoma složeno. Ne samo da je teško obezbediti da svi čvorovi istovremeno imaju istu informaciju, već je potrebno (relativno) mnogo vremena za skladištenje i izračunavanje. Ako su čvorovi smešteni na različitim usmeravajućim informacijama, ovaj koncept vodi do usmeravajućih zapetljavanja i teških odstupanja. Rezultat je mnogo zakrpa na protokolu, što prkosi standardu protokola kako bi se učinio pogodnijim (Mes 10).

Programeri B.A.T.M.A.N.-a su želeli čist početak. Odlučili su, između ostalog, da svaki čvor treba da poznaje samo sledeći skok, odnosno link lokalnog komšiju koji je staza između samog sebe i destinacije. Ono što su na mnogo načina uradili jeste da su učinili protokol jednostavnijim i lakšim za razumevanje. Na primer, način na koji B.A.T.M.A.N. izračunava optimalnu rutu, odnosno sledeći skok jeste upoređivanjem broja usmeravajućih poruka koje prima od svakog čvora i određivanjem ko je bio poslednji pošiljalac.

Usmeravajuće poruke poslate u B.A.T.M.A.N.-u nazivaju se Originator poruke (OGM). OGM format se promenio otkako je objavljen nacrt specifikacije (NALW10), ali još ne postoji zvanična publikacija s novim formatom paketa. Ažurirani paket format može se naći u unutrašnjoj dokumentaciji projekta. Paket formata pronađen u draft specifikaciji pripada starijoj verziji III B.A.T.M.A.N. algoritma. Algoritam korišćen u ovom radu je verzija IV (Slika 5.1).

Prava radna mašina paketa je polje Originator adresa, koju nosi: B.A.T.M.A.N. originator poruka (OGM) paket format. Adresa čvora ’A’ koji je emitovao OGM. Kada čvor ’B’ primi ovu poruku, on proverava da li su originator adresa i izvorna adresa IP zaglavlja iste – ako jesu, onda su dva čvora direktni susedi. B onda prosleđuje OGM menjajući samo ’Time To Live’ (TTL) i polja ’prethodnog pošiljaoca’. B-ijevi susedi

koji dobijaju ovaj OGM od A kroz B takođe prosleđuju paket i tako dalje. Sve OGM izvan B.A.T.M.A.N. mreže se emituju i reemituju dok TTL ne spadne na nulu, ili dok čvorovi ne prime OGM koji su prethodno sami poslali (Slika 8.1). Na ovaj način, sve OGM će biti primljene i reemitovane i svi čvorovi u mreži će čuti o međusobnom postojanju i koji su čvorovi prvi između njih i drugih čvorova, odnosno prvi skok staze. Svi čvorovi i njihovi prvi skokovi na njihovim putanjama čuvaju se na listi koja se naziva Originator list.

Slika 5.1 prikazuje kako se OGM emituje i prosleđuje kroz mrežu. Paket potiče od najlevljeg čvora i za svaki čvor koji prima paket, oni ga forvarduju svom komšiji ponovo.

Kad čvor koji je već primio i forvardovao OGM primi isti OGM od drugog čvora na kasnijem stupnju, on otpusti taj paket, tako da mreža neće biti preplavljena forvardovanjem iste OGM dok TTL ne bude nula. Ovo je takođe neophodno da bi se sprečilo usmeravanje petlji. B.A.T.M.A.N. koristi klizeće prozore kako bi otkrio da li je OGM ranije primljen ili nije.

5.1.2. B.A.T.M.A.N. Daemon u odnosu na B.A.T.M.A.N. Advanced

Postoje dve potpuno različite verzije B.A.T.M.A.N. ad hok usmeravajućeg protokola i onaj koji je do sada opisan je B.A.T.M.A.N. Daemon ili **batmand.** Ova verzija je korisna jer je mrežni sloj protokola za šemu provere identiteta predložen u ovom radu. Međutim, postoji nova verzija pod nazivom B.A.T.M.A.N. Advanced ili **batman-adv**, koja funkcioniše u link sloju. Ova verzija razbija standardni princip nanošenja slojeva jer on usmerava pakete kroz mrežu na link sloju, enkapsulirajući sve iznad kao IP i DHCP pakete.

B.A.T.M.A.N (batman-adv) je implementacija istoimenog protokola za rutiranje integrisanog u jezgru operativnog sistema Linux na sloju 2 OSI referentnog modela.

Većina drugih implementacija protokola za rutiranje (npr. OLSR, B.A.T.M.A.N.) radi na sloju 3 OSI referentnog modela, što znači da se informacije o rutiranju razmenjuju putem UDP paketa. batman-adv radi u potpunosti na 2 nivou OSI referentnog modela, što znači da se Eternet frejmovima transportuju i informacije o rutiranju, ali i podaci koji se prenose između čvorova. Sažima i prosleđuje celokupan saobraćaj po ugledu na virtuelni mrežni svič. Prednost je da čvorovi ne moraju biti svesni topologije mreže kao u promena u mreži.

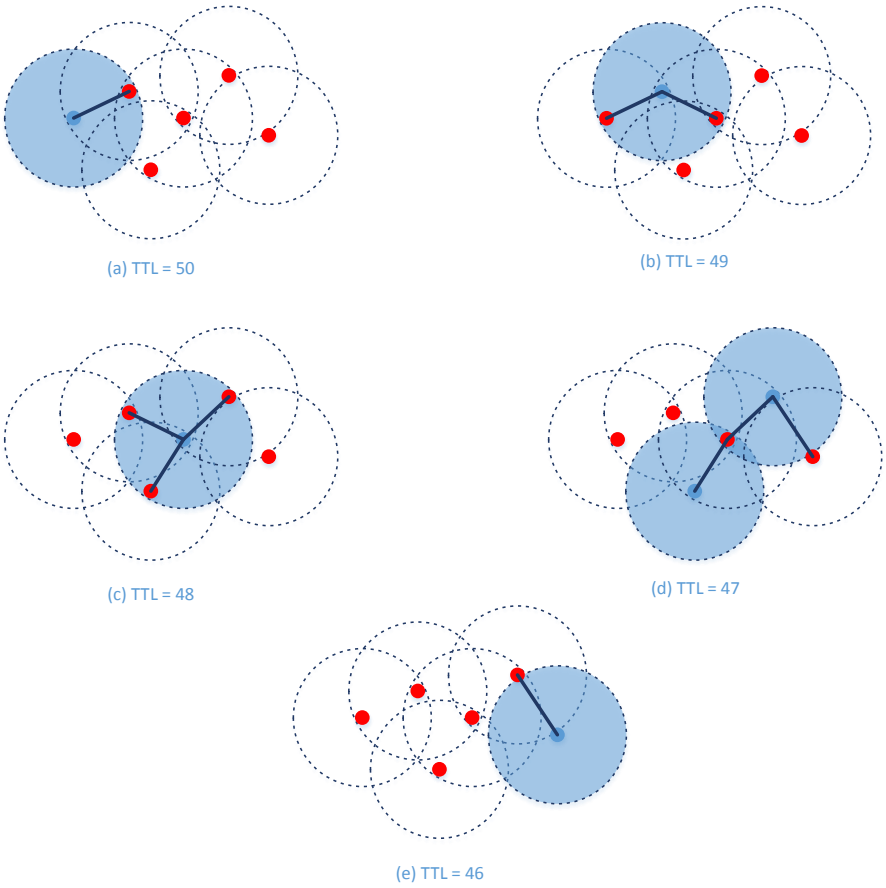
Ovakav dizajn ima zanimljive karakteristike:

- Mrežni sloj agnostika – možemo pokrenuti bilo koji protokol na vrhu B.A.T.M.A.N.-adv: IPv4, IPv6, DHCP, IPX...

- Čvorovi mogu učestvovati u mesh mreži bez prethodno dodeljene IP adrese.
- Laka integracija ne-mesh (mobilnih) klijenata (nije potrebno unositi obavezan HNA ručno).
- Roming ne-mesh klijenata.
- Optimizacija protokola podataka kroz mesh mrežu (alternativni interfejs, multikast, prenos korekcije greške itd.).
- Podržava protokole koji se oslanjaju na emitovanje mesh ili ne-mesh klijenata (Windows neighborhood, mDNS, striming itd.).

Version	Flags	TTL	GW Flag
Seq Nr.		GW Port	
Originator Address			
Previous Sender			
TQ	HNA Length		

Tabela 5.1 batman OGM format paketa



Slika 5.1 Tok jedne OGM poruke poreklom iz levog skupa čvorova

5.1.3. Implementacija B.A.T.M.A.N.-adv na OpenWRT-u

OpenWRT se najčešće opisuje kao linux distribucija za embeded uređaje. Prednost OpenWRT-a jeste da nije statički firmver, već podržava potpuno vrajtabl fajl sistem s upravljanjem paketima. Ovakav pristup oslobađa od selekcije aplikacija ili konfiguracije, što nam omogućava prilagođavanje uređaja za određenu aplikaciju odnosno funkciju. OpenWRT stvara okvir programeru da izgradi aplikaciju bez potrebe da izgradi kompletan firmver za nju. Korisnicima takođe omogućava prilagođavanje uređaja različitim potrebama.

Za čvorove WMN mreže u ovoj implementaciji koristimo WiFi uređaj TP-Link TL-WR741ND v4.x sledećih karakteristika:

```
system type             : Atheros AR9330 rev 1
  machine                 : TP-LINK TL-WR741ND v4
  processor               : 0
  cpu model               : MIPS 24Kc V7.4
```

Nakon instalacije OpenWRT firmvera, slede osnovna podešavanja uređaja:

Osnovna konfiguracija interfejsa:

```
less /etc/config/network
config interface 'lan'
        option ifname 'eth0'
        option type 'bridge'
        option proto 'static'
        option netmask '255.255.255.0'
        option ipaddr '192.168.130.1'
config interface 'wan'
        option ifname 'eth1'
        option proto 'dhcp'
config switch
        option name 'eth0'
        option reset '1'
        option enable_vlan '1'
config switch_vlan
```

```
        option device 'eth0'
        option vlan '1'
        option ports '0 1 2 3 4'
config interface 'adhocif'
        option _orig_ifname 'radio1.network1'
        option _orig_bridge 'false'
        option proto 'static'
        option ipaddr '10.1.1.3'
        option netmask '255.255.255.0'
/etc/config/wireless
config wifi-device 'radio0'
        option type 'mac80211'
        option channel '6'
        option macaddr 'f8:1a:67:a0:94:f2'
        option hwmode '11ng'
        option htmode 'HT20'
        list ht_capab 'SHORT-GI-20'
        list ht_capab 'SHORT-GI-40'
        list ht_capab 'RX-STBC1'
        list ht_capab 'DSSS_CCK-40'
        option txpower '27'
        option country 'US'
config wifi-device 'radio1'
        option type 'mac80211'
        option channel '6'
        option macaddr 'f8:1a:67:a0:c1:78'
        list ht_capab 'SHORT-GI-20'
        list ht_capab 'SHORT-GI-40'
        list ht_capab 'RX-STBC1'
        list ht_capab 'DSSS_CCK-40'
        option txpower '27'
        option country 'US'
```

```
config wifi-iface
        option device 'radio1'
        option encryption 'none'
        option ssid 'adhoc-mesh'
        option mode 'adhoc'
        option network 'adhocif'
config wifi-iface
        option device 'radio1'
        option mode 'ap'
        option ssid 'OpenWrt3'
        option network 'lan'
        option encryption 'psk-mixed'
        option key 'XXXXXXXX'
config wifi-iface
        option device 'radio1'
        option mode 'ap'
        option ssid 'Hotspot'
        option network 'hotspot'
        option encryption 'psk2'
  option key 'XXXXXXXX'
```

Da bismo instalirali B.A.T.M.A.N. Advanced nakon konfiguracije čvora, potrebno je sledeće:

```
root@OpenWrt:~#opkg update
root@OpenWrt:~#opkg install kmod-B.A.T.M.A.N.-adv
root@OpenWrt:~#opkg remove firewall ###brisanje firewall-a
root@OpenWrt:~#opkg remove dnsmasq ###brisanje dnsmasq
root@OpenWrt:~#uci delete network.wan uci delete
root@OpenWrt:~#network.wan6 uci delete network.lan uci delete
root@OpenWrt:~#network.@switch_vlan[0] uci delete
root@OpenWrt:~#wireless.@wifi-iface[0] uci set
root@OpenWrt:~#system.@system[0].hostname=batruter202 uci set
root@OpenWrt:~#network.@switch[0].enable_vlan=0 uci set
root@OpenWrt:~#network.mesh=interface uci set
```

```
root@OpenWrt:~#network.mesh.ifname=adhoc0 uci set
root@OpenWrt:~#network.mesh.mtu=1528 uci set
root@OpenWrt:~#network.mesh.proto=batadv uci set
root@OpenWrt:~#network.mesh.mesh=bat0 uci set
root@OpenWrt:~#wireless.radio0.disabled=0 uci set
root@OpenWrt:~#wireless.radio0.channel=7 uci set
root@OpenWrt:~#wireless.radio0.hwmode=11n uci set
root@OpenWrt:~#wireless.radio0.txpower=20 uci set
root@OpenWrt:~#wireless.wmesh=wifi-iface uci set
root@OpenWrt:~#wireless.wmesh.device=radio0 uci set
root@OpenWrt:~#wireless.wmesh.encryption=none uci set
root@OpenWrt:~#wireless.wmesh.ifname=adhoc0 uci set
root@OpenWrt:~#wireless.wmesh.network=mesh uci set
root@OpenWrt:~#wireless.wmesh.mode=adhoc uci set
root@OpenWrt:~#wireless.wmesh.ssid=A0:F3:C1:BB:78:AC uci set
root@OpenWrt:~#wireless.wmesh.bssid=A0:F3:C1:BB:78:AC uci set
root@OpenWrt:~#wireless.wmesh.mcast_rate=11000 uci commit
Sada editujemo /etc/rc.local:
root@OpenWrt:~#vi /etc/rc.local
################################################################
batctl bl 1
batctl f 0
brctl addbr br-bri
brctl addif br-bri eth0
ifconfig br-bri add 192.168.19.201
ifconfig eth0 up
ifconfig bat0 up
brctl addif br-bri bat0
ifconfig br-bri up
################################################################
root@OpenWrt:~#reboot
```

Na identičan način je konfigurisano 10 čvorova sa IP adresama:

Čvor0 192.168.19.200

Čvor1 192.168.19.201

Čvor2 192.168.19.202

Čvor3 192.168.19.203

Čvor4 192.168.19.204

Čvor5 192.168.19.205

Čvor6 192.168.19.206

Čvor7 192.168.19.207

Čvor8 192.168.19.208

Čvor9 192.168.19.209

batman-adv je WMN protokol koji radi na drugom sloju OSI referentnog modela, što je ranije opisano. Na ovaj način čvorovi su povezani u jedan kolizioni domen, što omogućava lakše upravljanje i veću bezbedonst mreže. Takođe, implementacija u ovakvoj konfiguraciji omogućava potpunu mobilnost čvorova, kao i lako i jednostavno proširenje mreže dodavanjem novih čvorova.

5.2. Pregled OLSR protokola

Prvobitno razvijen za mobilne ad hok mreže (MANET), danas se OLSR protokol često koristi u WMN mrežama**.**

Sledi kratak pregled osnovnog funkcionisanja OLSR protokola.

OLSR koristi koncept *Multipoint Relays* (MPR) za prosleđivanje kontrolnog saobraćaja kroz mrežu. MPR su podskup suseda jednog čvora koji imaju dvosmernu vezu ka njemu samom. Svaki set MPR čvora mora takođe da ima dvosmernu vezu sa susedima dva skoka od tog čvora. MPR se koristi kao srednji čvor za izračunavanje puta od izvora do odredišta (Jacquet et al., 2001). Ovo predstavlja prednost zato što se izbegava transport paketa na jednosmernim vezama. Važno za OLSR jeste da su MPR jedan hop susedi koji reemituju info poruke tog čvora; drugi jedan hop susedi će primiti i obraditi poruku, ali je neće reemitovati. Koristeći podskup čvorova smanjuje se broj poruka, a podaci se prenose čistije, ovo smanjuje mogućnost za preplavljivanjem mreže porukama (slika 8.1 i 8.2).

MPR se bira pomoću HELLO poruka koje emituje svaki čvor, one sadrže:

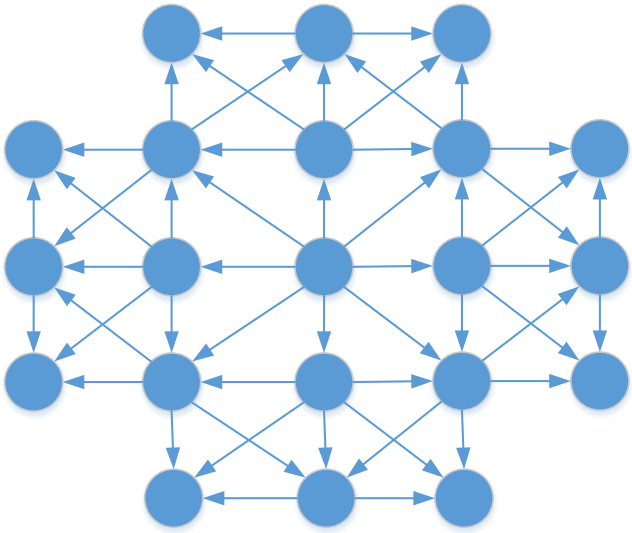
- spisak adresa suseda prema kojima postoji dvosmerna veza
- spisak adresa suseda koji mogu da osluškuju HELLO poruke.

5.2.1. Višestruki releji (MPR)

OLSR koristi koncept višestrukih releja (MPR) za prosleđivanje kontrole saobraćaja kroz mrežu. Ovi MPR su podskup jednoskoknih susednih čvorova i imaju dvosmernu

vezu s tim čvorom. Skup MPR-ova čvora mora takođe da ima dvosmernu vezu s čitavim dvoskočnim susedstvom tog čvora.

MPR se koriste kao srednji čvorovi za izračunavanje staze od izvora do destinacije. Ovo ima prednosti pri izbegavanju problema povezanih s prenosom paketa na neusmerenim linkovima. Važno je za OLSR da su MPR-ovi samo jednoskočni susedi čvora koji će retransmitovati emitovane poruke tog čvora; drugi jednoskočni sused će primiti i obraditi poruku, ali je neće



Slika 5.2 OLSR: Normalan protok bez MPR selekcije

retransmitovati. Korišćenjem podskupa čvorova koji prosleđuju saobraćaj jasno se smanjuje broj poruka koje se bočno kreću u mreži i tako smanjuju poruku. Slike 5.2 i 5.3 pokazuju razliku u smanjenom protoku izborom MPR, gde je potonji izabrao četiri plava čvora kao MPR.

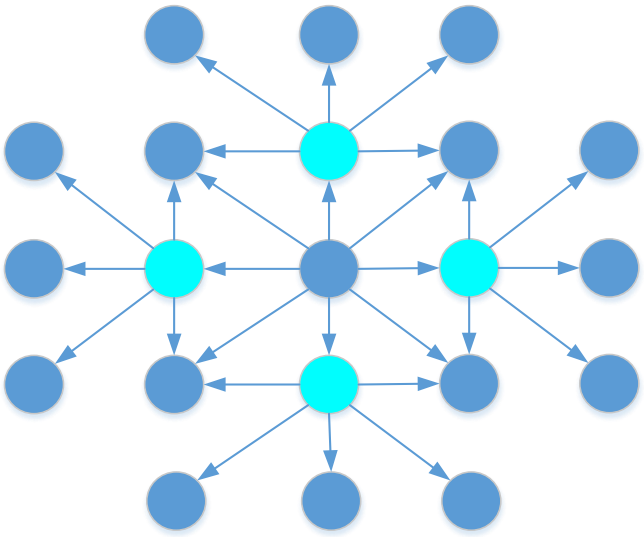
5.2.2. MPR selekcija

MPR se selektuju uz pomoć HELLO poruka koje emituje svaki čvor i sadrže:

- Spisak adresa suseda kod kojih postoji dvosmerni link
- Spisak adresa suseda koji mogu da čuju ovu HELLO poruku

Ako čvor primeti sopstvenu adresu na spisku adresa suseda koji mogu da čuju HELLO poruku, on smatra da je veza dvosmerna. Kad određeno vreme ne primi nijednu HELLO poruku, veza se vraća na jednosmernu. Ceo ovaj proces naziva se očitavanje linka.

Pomoću detekcije linka, čvorovi saznaju o jednosmernim i dvosmernim linkovima; sa ovim saznanjem čvorovi mogu izabrati svoj MPR skup od jednoskočnih suseda. Kad čvor izabere suseda kao MPR, to se takođe indicira u link statusu HELLO poruka.



Slika 5.3 OLSR: Smanjen protok sa MPR selekcijom

5.2.3. Usmeravanje

Pomoću HELLO poruka, čvor može da izgradi dve tabele: susedovu tabelu i MPR selektor tabelu. Kod susedne tabele, sve informacije jednoskočnog suseda, zajedno s primljenim HELLO porukama se kombinuju. MPR selektor tabela indicira koji čvorovi su izabrali čvor kao svoj MPR (kao što je indicirano link statusom u HELLO porukama). Topološka kontrola (TC) poruke, koja sadrži ovu MPR selektor tabelu, koristi se da izgradi usmeravajuću tabelu i distribuciju kroz mrežu, ali ih prosleđuje samo MPR. Čvor koji nije selektovan kao MPR ne emituje TC poruke. Samo spominjanjem MPR selektor seta, a ne svih suseda u kontroli saobraćaja, zaglavlje na kontrolnom paketu se smanjuje u OLSR.

TC poruke omogućavaju čvorovima da izgrade topološku tabelu u kojoj je zabeležena topologija mreže čuvanjem MPR-ova drugih čvorova u mreži (ovo je dovoljno jer se MPR-ovi koriste kao srednji čvorovi na stazi od izvora do destinacije). Upotrebom ove topološke tabele pune mreže, usmeravajuća tabela može se izračunati kao što je objašnjeno u Clausen et al., 2003. i može se odrediti naredni najbolji skok do destinacije.

5.2.4. Fish-eye ekstenzija

Link-state protokoli, kao OLSR osetljivi su na usmeravanje petlji, posebno u velikim mrežama. Kad mrežne topologije koje drže individualni čvorovi nisu sinhronizovane zbog gubitka paketa (u slučaju OLSR, to su TC paketi), usmeravajuće petlje mogu da postoje. OLSR koristi Fish-eye ekstenzije da spreči da te petlje postoje. To takođe poboljšava skalabilnost u usmeravajućem protokolu pošto će ekstenzija učiniti OLSR održivijim za velike i guste mreže kao što je prikazano u Barolli et al., 2009, zbog

manjeg zaglavlja proizvedenog TC porukama. Na osnovu nalaza da se usmeravajuće petlje više pojavljuju kod obližnjih čvorova nego kod udaljenih, princip ekstenzije Fish-eye je jednostavan: topološke informacije treba da se osveže više kod bliskih čvorova nego kod udaljenih.

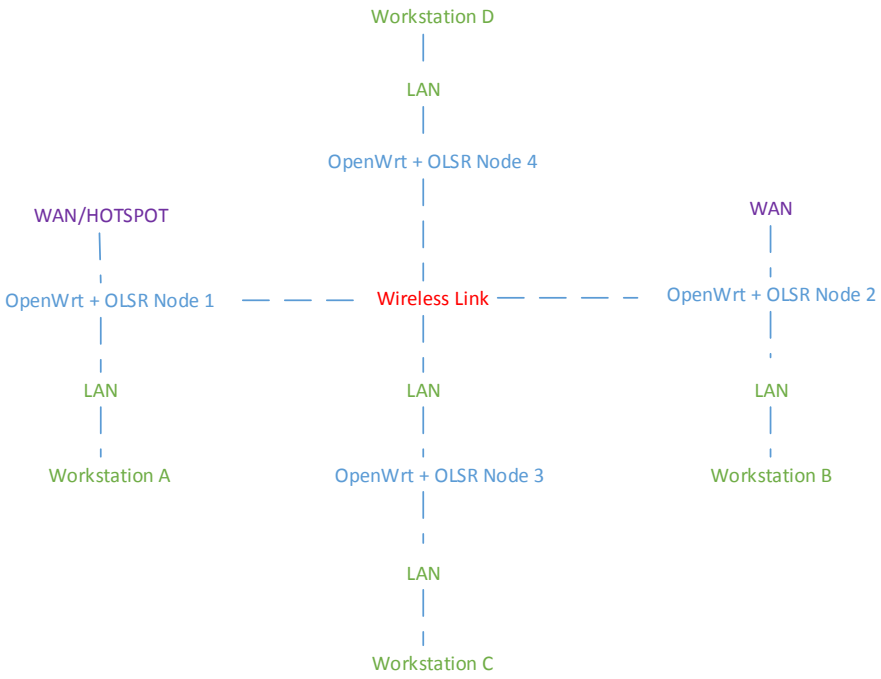
Dopuštanjem MPR da odluči o Time-To-Live (TTL) vrednosti njihovih TC paketa, MPR mogu da odrede frekvenciju kojom obližnji i udaljeni čvorovi primaju TC poruke. Sekvenca 2,8,2,16,2,8,2 i 255 TTL vrednosti se koriste u osnovi kod OLSR, tako da se samo svaki osmi TC šalje svim mrežnim čvorovima.

5.2.5. Implementacija OLSR

Karakteristika mesh mreža je samoorganizacija i automatska konfiguracija na osnovu promena topologije. Primer je da će se pravilno konfigurisan OLSR u mreži automatski organizovati u slučaju pada nekog od čvorova, ili kad se novi čvor pojavi na mreži, odnosno tada se dodaju ili brišu nove rute. OLSR je jedan od protokola za rutiranje kreiran za MANET mreže.

Postoje već razvijene varijante OpenWRT-a koje sadrže OLSRd primer je Freifunk projekat, međutim, u ovom slučaju je uzet izvorni OpenWRT firmver kome se dodaju moduli OLSRd.

Za čvorove WMN mreže u ovoj implementaciji koristimo WiFi uređaj TP-Link TL-WR741ND v4.x. Mrežu čvorova čine četiri uređaja s embeded linux distribucijom OpenWRT (Slika 5.4).

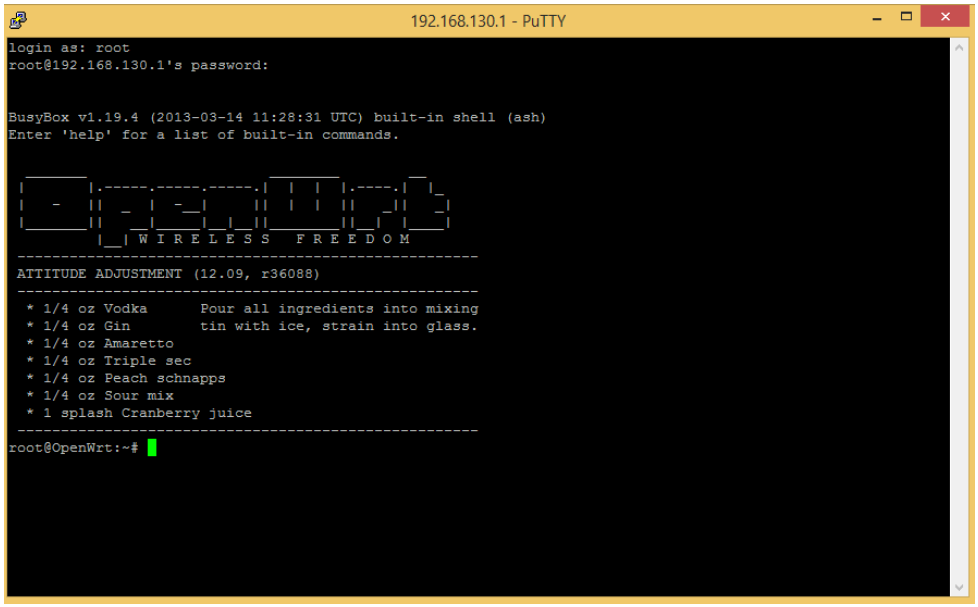


Slika 5.4 Konfiguracija WMN mreže sa četiri čvora

Nakon instalacije OpenWRT firmvera na uređaju sledi dodavanje modula za određeni protokol rutiranja, u ovom slučaju *olsrd*:

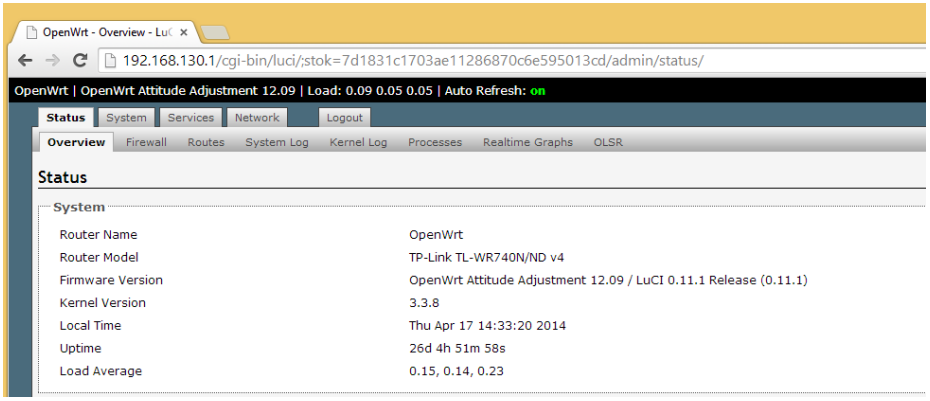
```
root@OpenWrt:~#opkg update

root@OpenWrt:~#opkg install olsrd olsrd-mod-arprefresh olsrd-mod bmf
       olsrd-mod-dot-draw  olsrd-mod-dyn-gw  olsrd-mod-dyn-gw-plain
       olsrd-mod-httpinfo   olsrd-mod-mdns   olsrd-mod-nameservice
       olsrd-mod-p2pd olsrd-mod-pgraph olsrd-mod-secure olsrd-mod-
       txtinfo  olsrd-mod-watchdog  luci-app-olsr  luci-app-olsr-
       services luci-app-olsr-viz
```

Sledi konfiguracija OLSRd čvorova (Slika 5.5).



Slika 5.5 CLI OpenWRT čvora

Konfiguraciju interfejsa na samom uređaju moguće je odraditi pomoću web interfejsa (Slika 5.6)



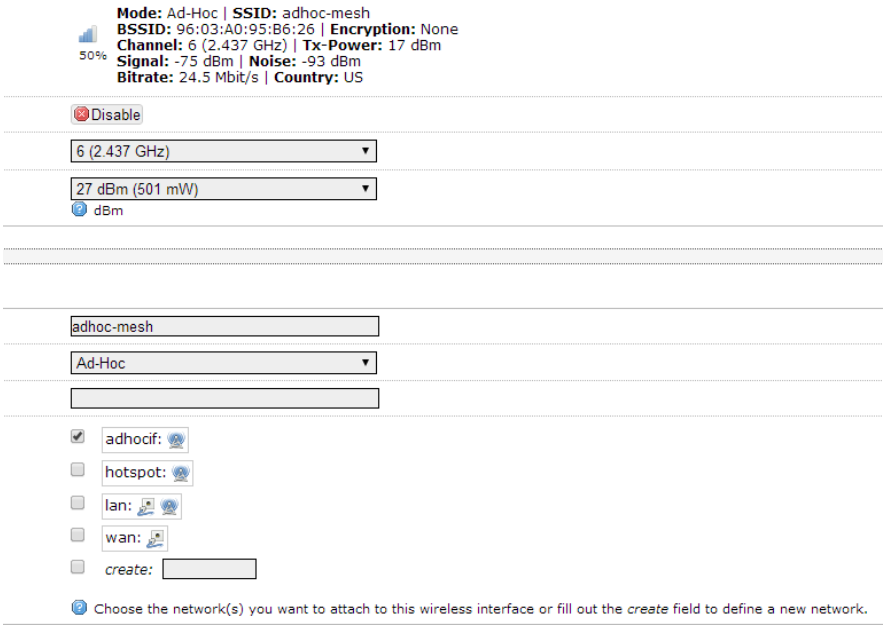
Slika 5.6 WEB interfejs OpenWRT čvora

Ili editovanjem fajla ‘/etc/config/wireless’ i ‘/etc/config/network’

```
/etc/config/wireless:
config wifi-device 'radio0'
        option type 'mac80211'
        option channel '6'
        option macaddr 'f8:1a:67:a0:94:f2'
        option hwmode '11ng'
        option htmode 'HT20'
        list ht_capab 'SHORT-GI-20'
        list ht_capab 'SHORT-GI-40'
        list ht_capab 'RX-STBC1'
        list ht_capab 'DSSS_CCK-40'
        option txpower '27'
        option country 'US'
config wifi-device 'radio1'
        option type 'mac80211'
        option channel '6'
        option macaddr 'f8:1a:67:a0:c1:78'
        list ht_capab 'SHORT-GI-20'
        list ht_capab 'SHORT-GI-40'
        list ht_capab 'RX-STBC1'
        list ht_capab 'DSSS_CCK-40'
        option txpower '27'
        option country 'US'
config wifi-iface
        option device 'radio1'
        option encryption 'none'
        option ssid 'adhoc-mesh'
        option mode 'adhoc'
        option network 'adhocif'
```

```
config wifi-iface
        option device 'radio1'
option mode 'ap'


/etc/config/network:
config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'
config interface 'lan'
        option ifname 'eth0'
        option type 'bridge'
        option proto 'static'
        option netmask '255.255.255.0'
        option ipaddr '192.168.130.1'
config interface 'wan'
        option ifname 'eth1'
        option proto 'dhcp'
config switch
        option name 'eth0'
        option reset '1'
        option enable_vlan '1'
config switch_vlan
        option device 'eth0'
        option vlan '1'
        option ports '0 1 2 3 4'
config interface 'adhocif'
        option _orig_ifname 'radio1.network1'
        option _orig_bridge 'false'
```

Prednost uređaja i samog OpenWRT softvera koji je izabran jeste taj što drajveri za WLAN interfejs imaju podršku za konfiguraciju virtuelnih interfejsa. To znači da je moguće konfigurisati ad hok linkove što predstavlja jedan virtuelni interfejs, a u isto vreme drugi virtuelni interfejs obavlja funkciju bazne stanice ili hotspota. Kičma WMN mreže oslanja se na ad hok linkove, dodavanje novih čvorova je veoma lako uz pravilnu konfiguraciju.



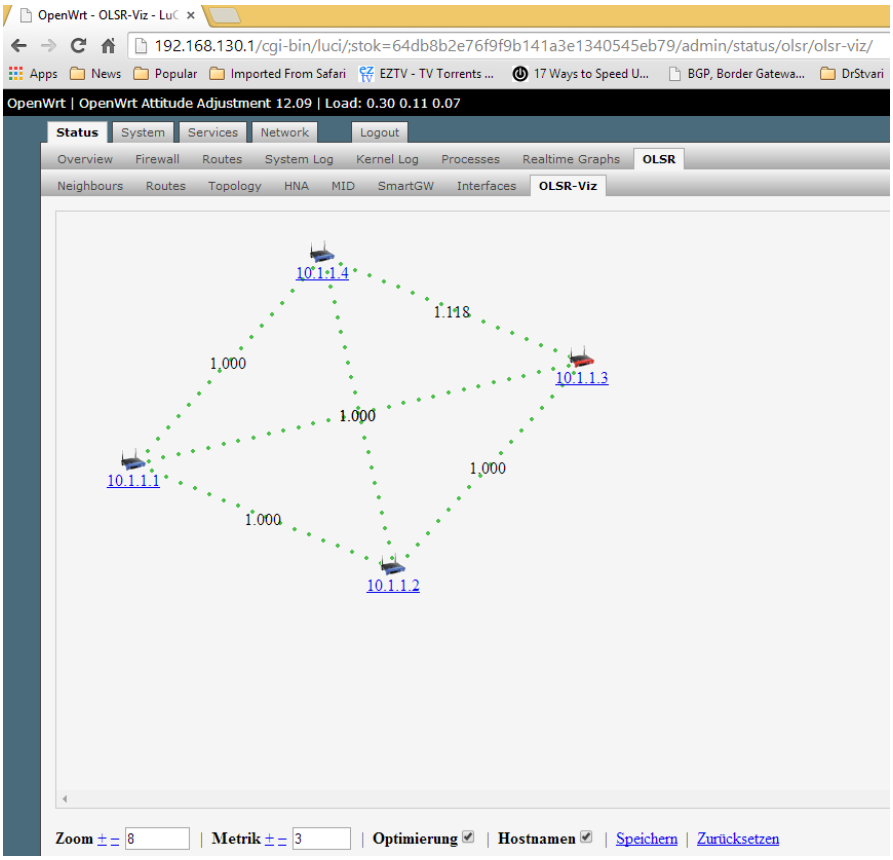
Slika 5.7 Konfiguracija interfejsa

Posle konfiguracije interfejsa (Slika 5.7), sledi podešavanje OLSRd koji je prethodno instaliran. U fajlu */etc/config/olsrd* nalaze se osnovni parametri konfiguracije:

```
config olsrd
        option IpVersion '4'
config LoadPlugin
        option library 'olsrd_arprefresh.so.0.1'
        option ignore '0'
config LoadPlugin
        option library 'olsrd_httpinfo.so.0.1'
        option port '1978'
        list Net '0.0.0.0 0.0.0.0'
        option ignore '0'
```

```
config LoadPlugin
        option library 'olsrd_nameservice.so.0.3'
        option ignore '0'
config LoadPlugin
        option library 'olsrd_txtinfo.so.0.1'
        option accept '0.0.0.0'
        option ignore '0'
config Interface
        option ignore '0'
        option interface 'adhocif'
        option Mode 'mesh'
config InterfaceDefaults

config LoadPlugin
        option library 'olsrd_dyn_gw_plain.so.0.4'
        option ignore '0'
config LoadPlugin
        option library 'olsrd_dot_draw.so.0.3'
        option ignore '1'
config LoadPlugin
        option library 'olsrd_mdns.so.1.0.0'
        option ignore '0'
config Hna4
        option netmask '255.255.255.0'
        option netaddr '192.168.130.0'
```

Nakon konfiguracije svakog čvora, mesh mreža će izgledati kao na Slici 5.8.



Slika 5.8 Vizuelni prikaz čvorova

5.3. Pregled rezultata istraživanja

Mobilne ad hok mreže imaju sposobnost stvaranja dinamičnih promena u bilo kojoj topologiji. Karakteristike po kojima se razlikuju MANET-i od drugih mreža jeste da su sposobni da menjaju svoju lokaciju. Rezultati istraživanja u poslednjih pet godina pokazali su da oni mogu prevazići kongestiju i komunikacione barijere između dva čvora. Ova studija upoređuje performanse dva različita protokola rutiranja u ad hok mreži s multiskokovima: Better Approach To Mobile Ad hoc Networks (B.A.T.M.A.N) i Optimized Link State Routing Algorithm (OLSR). Ovi protokoli su osmišljeni da maksimizuju dostavu količine paketa, propusnu moć, da minimizuju end-to-end kašnjenje i za učitavanje ruta za poboljšanje performansi. Ovaj dokument proučava link-state, distance-vector pristupe rutiranju koristeći OLSR i B.A.T.M.A.N. protokole rutiranja koji su razmatrani u ranijim radovima i na realnom testiranju u opisanim uslovima iz prethodnog poglavlja. Dokument zaključuje da B.A.T.M.A.N. po performansama prevazilazi OLSR u izrazima bolje propusne moći, manjeg kašnjenja, manjeg opterećenja procesora (CPU) i manjih opštih troškova rutiranja kad se uzme u obzir eksperimentalna procena. Ali, kad se uzmu u obzir drugi pristupi, vidi se da su slični po svojim performansama.

Rutiranje je centralno i jedno od najvažnijih područja bežične multi-hop ad hok mrežne arhitekture. Ali, samo nekoliko pravih svetskih eksperimentalnih studija istraživalo je WMN rutiranje, uprkos njegovoj važnosti i stotinama različitih protokola rutiranja predloženih tokom poslednje decenije. Različiti protokoli rutiranja bili su predloženi da unaprede performanse. Svi protokoli rutiranja ponašaju se drugačije nego drugi u poboljšanju i održavanju performansi rutiranja. U rutiranju postoje dva tipa protokola: reaktivni (npr. DSR) i proaktivni (npr. OLSR, B.A.T.M.A.N.).

Ovaj rad proučava eksperimentalno poređenje Optimized Link State Routing (OLSR) i Better Approach To Mobile Ad hoc Networking (B.A.T.M.A.N.), izvedeno realnim eksperimentima i drugih istraživača. Ovi protokoli predstavljaju različite pristupe rutiranju u multihop i ad hok mrežama.

5.3.1. Protokoli rutiranja

Postoje dva tipa protokola rutiranja:

1. Protokol rutiranja na zahtev (reaktivni)

Ovaj protokol nalazi rutu kako je zahtevano. Pre svega, protokol šalje paket zahteva za rutu svim prednjim čvorovima. Ovo se čini dok se ne nađe odredište. Odredište šalje paket odgovora za rutu izvornom čvoru. Samo se ruta odziva održava, ostalo se briše. Posle toga sve rute se održavaju u tabeli. Koristi se dinamično rutiranje izvora (Dynamic Source Routing – DSR). Konekcija se stvara između izvora i odredišta kad čvor izvora pošalje zahtev.

2. Protokol rutiranja pokrenut tabelom (proaktivni)

Ovaj protokol rutiranja pravi novu listu odredišta pre nego što se zahteva rezultat. Ruta se održava tabelom rutiranja. Izvor šalje paket sledećem čvoru čineći time promene u tabeli rutiranja. Razni proaktivni protokoli rutiranja su:

Better Approach To Mobile Ad Hoc Network (B.A.T.M.A.N.): (Kulla et al., 2010; Barolli et al., 2009; Kiess et al., 2007). Pre svega, šalje početnu poruku (OriGinator Message – OGM). Veličina OGM je 52 bajta. To sadrži informaciju o IP adresi čelnog čvora i broj sekvence se takođe uvećava. Zatim se uzima najbolji čvor koristeći njegov rang. Ako broj sekvenci čvora varira u okviru opsega, onda je konekcija dvosmerna.

Optimized Link State Routing algorithm (OLSR): (Barolli et al. 2009; Kulla et al., 2010; Clausen et al., 2003; Paul et al., 2011) OLSR proizvodi različite veze između čvorova kao što je dato od izvora do odredišta. Prvo šalje HELLO poruku da proveri svog suseda. Ovo stvara promene u tabeli rute posle svakog čvora, kad je moguć prenos paketa. Tabelom rutiranja rukovodi informacija Kontrole topologije (Topology control) (TC) paketa. Kontrolni paket šalje se mreži preko specijalnog čvora zvanog relej s više

tačaka. Usled toga, kontrolni saobraćaj je redukovan; putanja se bira korišćenjem algoritma najkraće putanje.

5.3.2. Izazovi u protokolima multihop ad hok mreža

Izazovi u protokolima multihop ad hok mreže su sledeći:

U hijerarhijskom rutiranju, ako jedan čvor zakaže, nova IP adresa i mrežna maska bili bi potrebni da oforme ponovo vezu s drugim ruterom. To je razlog zašto hijerarhijske adresirajuće postavke neće funkcionisati u multihop ad hok mrežama. Stoga adresirajuće strukture treba da budu indiferentne.

Multihop i ad hok čvorovi su često jeftine slabe mašine jer moraju da se izbore s mnoštvom spoljnih prilika. Zato je snaga procesora ovih uređaja ograničena.

Postoje konstantne promene u multihop ad hok mrežama (Jacquet et al., 2002) i stoga tradicionalni protokoli rutiranja kao što su RIP i OSPF ne mogu biti korišćeni pošto se ažuriraju previše retko. Čest tok HELLO poruka i razmene topologija potreban je da se prate stanja veze/linka koja su u konstantnoj promeni.

Pošto se stanja u jednoj ad hok mreži stalno menjaju, informacija rutiranja mora biti ne(po)vezano emitovana, u cilju održanja efikasnosti u sredini koja se (međusobno) deli. Režijski troškovi biće prema tome viši u protokolima ad hok rutiranja.

Prema Split Horizont pravilu, ako jedan interfejs prima neku rutu, onda ta ista ruta nikad ne treba da bude ponovo objavljena kroz isti interfejs. Ovo pravilo koristi se da bi se izbegla count-to-infinity petlja rutiranja u žičanim mrežama. U multihop ad hok mrežama, čvorovi moraju ponovo da pošalju informaciju rutiranja preko istog interfejsa, što znači da Split Horizont pravilo ne sme da se koristi. RIP i EIGRP su prema tome neprimenjivi.

Samoformirajuća i samooporavljajuća svojstva su veoma daleko od toga da nemaju sposobnost da manuelno kreiraju varijable kao što su propusni opseg i kašnjenje. Pošto takve karakteristike ostaju postojane u multihop ad hok mrežama, one ne mogu manuelno da unose promene kao što mogu da budu unesene u OSPF i EIGRP. Kao rezultat, metrike projektovanja rutiranja za multihop ad hok mreže postaju težak zadatak. Metrika brojanja skokova se koristi za pojednostavljenje tog procesa i ograničenja su dobro poznata. Tradicionalni problemi se pogoršavaju u multihop ad hok mrežama jer će putanje s manje skokova pre biti rute između udaljenih linkova s manjim prenosom podataka. U mnogim slučajevima to će voditi do korišćenja veće udaljenosti, s putanjama manjih brzina. Ovi nenamerne interakcije ukrštenih slojeva vodile su do pogoršanja kvaliteta performansi (Kawadia et al., 2005). Brojanje skokova se slabo izvodi u multihop ad hok mrežama (De Couto et al., 2005).

5.3.3. Metode prevazilaženja izazova

Različiti su načini prevazilaženja izazova s kojima se suočavaju multihop ad hok mreže:

- Ograničeno rasprostiranje

Popularan pristup da se redukuju opšti troškovi rutiranja (i u proaktivnim i u reaktivnim protokolima) jeste da se ograniči rasprostiranje informacija rutiranja. Primena ovog koncepta na link-stejt rutiranje poznato je kao stejt-rutiranje ribljeg oka (Fish-eye State Routing – FSR) (Pei et al., 2000). Pokazalo se da ove tehnike primetno redukuju opšte troškove (Macker et al., 2003). Razlog zašto neke od nepreciznih ili neznatno netačnih informacija mogu biti tolerisane jeste taj što se odluke rutiranja čine na skok-po-skok osnovi. To znači da ako je neki čvor udaljen mnogo skokova, onda će često biti dovoljna ruta u generalnom pravcu.

FSR modifikuje Time to Live (TTL) u porukama rutiranja da bi se ažurirali bliski i udaljeni čvorovi u različitim intervalima. Studije su pokazale da FSR pribavlja bolju optimizaciju u velikim mrežama s velikim dijametrom. Uključenje FSR u OLSR (Andreas et al., 2009) je doprinos njegovoj efektivnosti.

- Metrike rutiranja

ETX (ili brojanje/obračun očekivane transmisije (Expected Transmission Count) (Basagni et al., 1998) je metrika pouzdanosti namenjena pronalaženju putanja koje zahtevaju najmanje transmisija. Uprkos svim paketima primljenim korišćenjem zahteva za automatsko ponavljanje (Automatic Repeat Request) (ARQ) u 802.11, ponovljene transmisije rezultiraju gubljenjem vremena emitovanja i otuda propusne moći. ETX izračunava verovatnoću uspešnih transmisija u oba smera preko bežičnog linka. Da bi se odredile ove statistike, postoji emitovanje konfigurisanog broja probâ od svakog čvora. Prijemnici izračunavaju broj proba koje su primili i porede ih s očekivanim brojem. Pošto su linkovi simetrični, važno je da se meri količina uspešnih transmisija u oba smera. Da bi se pribavile ove informacije, svaki čvor postavlja svoje vlastite ETX vrednosti u probe koje šalje. Formula za izračunavanje ETX jednog linka pokazana je u jednačini 1. Postoje dobro dokumentovani problemi s ETX (Akyildiz et al., 2009). ETX ne inkorporiše propusnu moć i to izgleda da je najveći problem. Ovo može uzrokovati da ETX favorizuje manji broj sporih linkova veće udaljenosti na uštrb većeg broja linkova velike brzine. Uprkos ovim problemima, ETX (Johnson et al., 2008) se koristi u brojnim protokolima rutiranja kao što je OLSR (Clausen, T et al., 2003).

$$\text{ETX (jednog linka)} = N/P = 1/(LQ \times NLQ)$$

gde je N = ukupan broj transmisija bez greške potreban da se prenesu P paketi, P = ukupan broj paketa, LQ = deo uspešnih paketa koji su primljeni od suseda u okviru

perioda izloženosti (window period) i NLQ = deo uspešnih paketa koji su primljeni od susednog čvora u okviru perioda izloženosti.

Metrika očekivanog vremena transmisije (ETT) (T. Clausen et al., 2003) dodaje sposobnost merenja propusne moći, unapređujući time ETX. Implementacije ETT su ograničene jer zahtevaju standardizovan način pribavljanja količine podataka od bežičnog drajvera. Sve dok takvi mehanizmi ne budu široko rasprostranjeni, implementacije ETT biće problematične i imaće problem međuoperatibilnosti. ETT je značajno unapređenje u odnosu na ETX, ali praktična primena je teška.

5.3.4. Procena performansi

Da bi se procenila efikasnost performansi protokola rutiranja, razmatraju se sledeći parametri:

Koeficijent/količnik dostave paketa (PDR – Packet delivery ratio): količnik paketa poslatih s izvora s brojem paketa primljenih na odredištu. PDR se određuje ovako:

$$PDR = P_r / P_s$$

gde je Pr ukupan broj primljenih paketa, a Ps ukupan broj poslatih paketa. Veća vrednost količnika dostavljenih paketa znači bolje performanse protokola.

Kašnjenje s kraja na kraj (D_{avg}): prosečno vreme koje je potrebno jednom paketu podataka da stigne na odredište. Ono uključuje moguće kašnjenje uzrokovano baferovanjem tokom otkrivanja rute. To je kašnjenje paketa poslatog od izvora do odredišta. Prosečno kašnjenje izračunava se ovako:

$$(D_{avg}) = \Sigma (t_r - t_s) / \Sigma \text{ Broj konekcija}$$

gde je t_s vreme slanja paketa, a t_r vreme primanja paketa. Niža vrednost kašnjenja s kraja na kraj znači bolje performanse protokola.

Nosivost/opterećenje rutiranja {Routing Load (RL)}: opterećenje rutiranja je broj kontrolnih paketa rutiranja koji su preneseni za svaki paket podataka koji je dostavljen na odredište. Opterećenje rutiranja se određuje ovako:

$$RL = P_c / P_d$$

gde je P_c ukupni broj kontrolnih paketa koji je poslat, a P_d ukupan broj paketa koji je dostavljen.

Protok: ukupan broj paketa primljenih na odredište.

5.3.5. Analiza izveštaja

Nakon razmatranja sveukupnog istraživanja (Jayakumar et al., 2007), upoređujući ova dva različita protokola, u rezultatu se može utvrditi:

Kada se 100 čvorova s dužinom paketa od 50.000 bajta pošalje preko B.A.T.M.A.N.-a i OLSR-a, količnik dostavljenih paketa (PDR), kašnjenje s kraja na kraj, opterećenje rutiranja i propusna moć su bolji sa OLSR-om nego sa B.A.T.M.A.N.-om. Takođe, kad se uključi faktor mobilnosti (Mobilnost = 30m/s), opet je bio bolji OLSR.

Veoma je teško dizajnirati protokol rutiranja koji zadovoljava sve parametre i izabrati najbolji sa svih tačaka gledišta. Ali ovde smo (zasnovano na rezultatima analize istraživanja) uporedili ova dva protokola rutiranja.

5.3.6. Eksperimentalna studija

U ovoj eksperimentalnoj studiji batman-adv i OLSR-a (Murray et al., 2010) učinjeni su pokušaji da se koristi rutiranje ad hok na zahtev vektora distance (Ad hoc On demand Distance Vector – AODV), međutim, slično nedavnim studijama (Abolhasan et al., 2009), problemi s implementacijom učinili su to neizvodljivim.

Protokol rutiranja B.A.T.M.A.N. razvija se i kao protokol rutiranja korisničkog prostora (user-space), koji operiše u mrežnom sloju, isto kao i u implementaciji u prostoru kernela (kernel-space), u sloju veze podataka. Ova studija je urađena eksperimentima (Murray et al., 2010) vođenim s oba protokola rutiranja i na njih se odnose nazivi batman L3 i batman L2. Postoji samo nekoliko stvarnih svetskih eksperimentalnih procena ovih protokola (Johnson et al., 2008; Abolhasan et al., 2009).

Implementacija OLSR-a u Linuksu koju je razvio Tonnesen (Tønnesen 2004) bila je korišćena za poređenja u ovoj studiji. Ova implementacija označava se kao olsr.org. Ona je sada deo najveće inicijative otvorenog izvora (Open Source) ad hok mrežnog razvoja. Verzija 0.5.5 koja je usaglašena sa RFC3626 je korišćena i sposobna za korišćenje nove ETX metrike za izračunavanje optimalnih ruta isto kao i za korišćenje optimizovane verzije algoritma Dijkstra.

U eksperimentu predloženom u Johnson et al., 2008, bežični čvorovi bili su ALIX 500MHz x86 ugrađeni PC sa 256 MB RAM-a i Atheros CM9 bežičnim karticama. Platforma i protokol rutiranja mogu se naći na tabeli 5.2. Svi protokoli rutiranja bili su testirani s njihovim fabričkim prepodešenim konfiguracijama.

Platforma	Verzija	Protokol rutiranja	Verzija
Voyage Linux	0.6	OLSR	0.5.6-rc7
Linux Kernel	2.6.30-486-voyage	BAT L3	0.3
MadWiFi	0.9.4	BAT L2	0.2

Tabela 5.2 Platforma i konfiguracije rutiranja

Komparativni testovi su izvedeni u (Tønnesen 2004) četiri različite topologije. Prva topologija je postavljena stavljanjem svih čvorova u direktni komunikacioni opseg

veznog čvora. U ovoj topologiji nije se uspostavljalo nikakvo rutiranje i zato je korišćena kao kontrola.

Ostale tri topologije predstavljale su nasumično postavljanje čvorova kroz jednu zgradu u kojoj su se čvorovi držali dovoljno razdvojeno da bi se osigurala multihop topologija. U predloženoj eksperimentalnoj postavci u (Murray et al., 2010), transmisiona moć bila je redukovana i svi bežični čvorovi su bili postavljeni u različitim sobama. Parametri koji su mereni bili su količnik dostave paketa, propusna moć i opšti troškovi protokola rutiranja.

U testu propusne moći (Murray et al., 2010), jedan vezni čvor bio je konektovan na namenski server koji je radio na *lighthttpd* veb serveru. Bežičnim čvorovima su date instrukcije da preuzmu simultano veliki fajl od 158MB sa *lighthttpd* servera i preuzimanje je bilo vremenski određeno. Proteklo vreme od trenutka kad je izdata komanda do momenta kada je poslednji čvor završio transfer fajla bilo je zapisano. Ovi testovi bili su izvršeni više puta za svaki protokol rutiranja u svakoj topologiji. I, da zaključimo, izvršeno je izračunavanje prosečnih vrednosti ovih rezultata.

U toj studiji, opšti troškovi protokola rutiranja bili su takođe zabeleženi. Kad su rutirali hiljadu paketa u sekundi, čvorovi nisu bili dovoljno jaki da uhvate saobraćaj koji je prolazio kroz njihove interfejse. To je vodilo do teškoća u određivanju tačnih opštih troškova rutiranja. Opšti troškovi protokola rutiranja bili su mereni postavljanjem svih bežičnih čvorova unutar opsega eksternog uređaja za hvatanje saobraćaja. Korišćen je Wireshark da hvata pakete s intervalima dužim od 60 sekundi.

5.3.7. Rezultati i konačna analiza

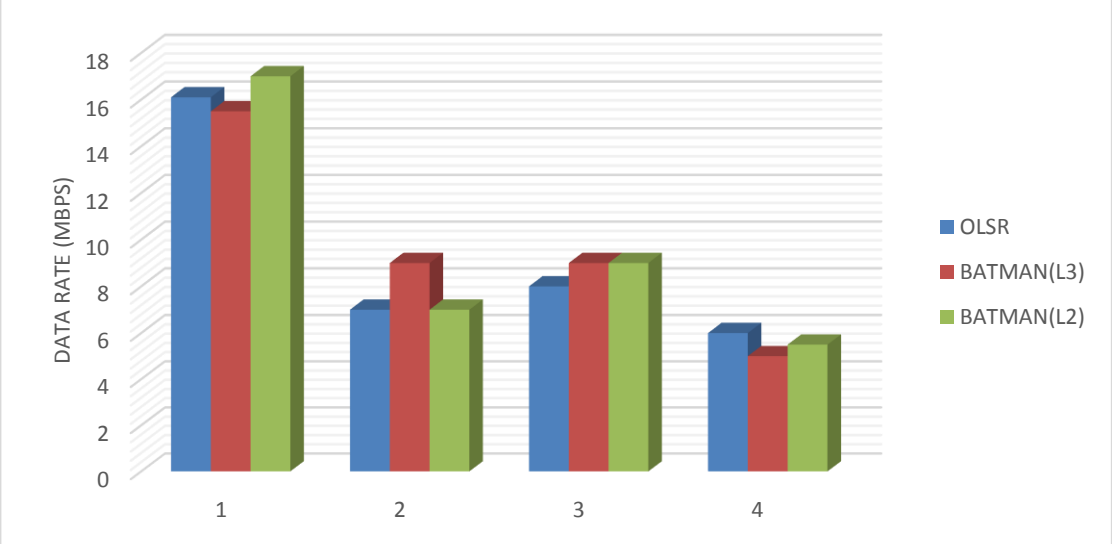
Različiti protokoli imaju različite performanse uz razne parametre. Različiti tipovi varijacija su urađeni sa čvorovima, kao što su varirajući broj čvorova, dužina paketa i mobilnost. Zasnovano na analizi istraživanja kad je uzet maksimalni broj čvorova s maksimalnom dužinom paketa, OLSR je bolje funkcionisao nego B.A.T.M.A.N. Takođe, kad je istom scenariju dodat maksimalni faktor mobilnosti, opet je OLSR bio bolji od batman-adv.

Efekat čvorova (N=100) i dužine paketa (50.000 bajta) na performanse je obrađen u analizi istraživanja. OLSR radi najbolje jer OLSR ima kraće kašnjenje. Budući da je proaktivni protokol, OLSR održava svežu listu čvorova. Tokom pristizanja paketa, on ih ili otprema ili napušta.

Efekat čvorova (N=100) i mobilnosti (30 (m/s)) na performanse je takođe obrađen u analizi istraživanja. OLSR najbolje funkcioniše kad je mobilnost visoka. B.A.T.M.A.N. ima dobre performanse kad mobilnost raste, ali su slabije od OLSR.

Zasnovano na eksperimentalnoj studiji (Murray et al., 2010), svi protokoli rutiranja bili su podjednako pouzdani i TCP ispadi podataka bili su veoma retki. Konzistentan količnik dostave paketa je posmatran za tri protokola rutiranja. Mada su svi količnici dostave paketa bili između 99,6% i 99,98%, rezultati su varirali za različite topologije. Ovi rezultati se razlikuju od drugih eksperimentalnih studija (Johnson et al., 2008; Abolhasan et al., 2009), koje nalaze značajno niže količnike dostave paketa jer su te studije vršene na koordinatnoj mreži, koja je bila izabrana kao logična topologija bežičnog okruženja za testiranje. Ona je odabrana zbog sposobnosti da kreira potpuno povezanu gustu sito mrežu i mogućnosti kreiranja mnoštva drugih topologija selektivno se premeštajući na posebne čvorove, i da čini da ponavljanje eksperimenta bude moguće.

Test propusnog opsega dao je sledeći grafikon:



Grafikon 5.1 Performanse različitih algoritama za razne topologije (Murray et al., 2010)

Grafikon 5.1 pokazuje da batman L2 ima bolje performanse od batmana L3 i OLSR-a u tri od četiri topologije. Ali, razlike u performansama su previše male da bi se moglo išta definitivno zaključiti. Studija (Johnson et al., 2008) koja upoređuje OLSR i B.A.T.M.A.N. našla je da je propusna moć B.A.T.M.A.N.-a približno 15% bolja od OLSR-a. Jedna druga studija (Murray et al., 2010) osporava taj fakt zasnovan na izboru varijabli mreže. Ona kaže da prepodešeno OLSR ima HELLO interval od 2 sekunde i interval od 5 sekundi za razmenu topologije. Komparativno, B.A.T.M.A.N. prenosi jednu u potpunosti drugačiju poruku poznatu kao OGM svake sekunde. U prethodnoj studiji (Johnson et al., 2008), radi pravednosti, HELLO intervali OLSR-a i razmene topologije održavani su isto kao OGM intervali B.A.T.M.A.N.-a od jedne sekunde. Ali, u kolegijalnoj studiji (Murray et al., 2010), ovo je nepravedno jer su B.A.T.M.A.N. i OLSR u potpunosti različiti protokoli. B.A.T.M.A.N.-ove OGM su veoma male jer nose

veoma malo informacija rutiranja i potrebno je da budu slate češće nego HELLO poruke i razmene topologije OLSR-a. Osporavanje koje je ovde prikazano je potvrđeno u studiji (Johnson et al., 2008) koja je izvedena a koja poređuje oba protokola veoma efikasno i stoga se njen značaj ne može poreći.

U ovom dokumentu, B.A.T.M.A.N. i OLSR protokoli su upoređeni zasnovano na izvedenim eksperimentalnim studijama (Johnson et al., 2008; Murray et al., 2010; Abolhasan et al., 2009), kao i na realnim testovima dve mesh mreže. Prvu mrežu sačinjava četiri čvora zasnovana na Tp-link WR740N i OLSRD. Druga mreža se sastoji od 10 TP-Link WR740N čvora sabatman-adv L2 ruting protokolom. Performanse ovih protokola analizirane su na osnovu međusobnog uticaja čvorova u mreži, dužine paketa (saobraćaj) i mobilnosti. Sveukupni rezultat analize istraživanja pokazuje da OLSR dobro funkcioniše u velikoj mreži (čvorovi = 100), kao i kad se uračuna faktor mobilnosti. Tako, u svim scenarijima, OLSR pokazuje bolje rezultate nego B.A.T.M.A.N. Eksperimentalni rezultati (Murray et al., 2010) potvrđuju da su opšti troškovi OLSR-a veći nego B.A.T.M.A.N.-a (Johnson et al., 2008; Abolhasan et al., 2009), ali su u kontradiktornosti s drugim studijama koje tvrde da postoje velike razlike u propusnom opsegu između OLSR-a i B.A.T.M.A.N.-a (Johnson et al., 2008; Abolhasan et al., 2009). Rezultat studije (Murray et al., 2010) sugeriše da su performanse OLSR-a i B.A.T.M.A.N.-a slične. Zato se ne može izvesti nikakav specifičan zaključak zasnovan na nekoj od istraživačkih studija (Johnson et al., 2008;Murray et al., 2010; Abolhasan et al., 2009) i zbog toga se moraju izvoditi dalji eksperimenti.

Ovo upoređenje protokola rutiranja za WMN mreže u realnim uslovima takođe daje vrlo slične rezultate u performansama. Ukoliko je reč o mrežama s manje od 60 čvorova, svi testovi su pokazali vrlo slične rezultate. Na osnovu analiza i realne konfiguracije može se zaključiti da primenu jednog od protokola za rutiranje, u ovom slučaju OLSRd ili B.A.T.M.A.N.-adv možemo izabrati u zavisnosti od servisa koje želimo da distribuiramo u mreži. Na primer, ako bi zadatak bio da korisnicima pružimo internet servis s jedne centralne lokacije i integrišemo L2 PPPoE protokol, preporuka *je batman-adv* koji se zasniva na L2 rutiranju. U slučaju ako imamo potrebu da delimo mrežne resurse sa više statičkih WMN čvorova koji bi činili okosnicu mreže, preporuka je OLSRd koji se zasniva na L3 rutiranju.

Na osnovu analize ova dva protokola zaključuje se da su potrebni dalji eksperimenti i analize, kao i optimizacija WMN protokola za mesh mreže.

6. EMULATOR ZA ANALIZU WMN PROTOKOLA – MESHLAB

6.1. Emulatori

Emulator predstavlja virtuelnu mašinu koja simulira kompletan hardver, što omogućava gost OS-u da se izvršava na potpuno nezavisnom procesoru. Emulatori predstavljaju dobro rešenje za testiranje novih uređaja, softvera ili protokola. Emulator je sposoban da emulira OS, ili neki uređaj (Android, WiFi ruter i dr.), u ovom slučaju, mesh čvor koji podržava OLSR protokol rutiranja.

6.2. Prednosti emulatora u odnosu na simulatore

Mrežni simulatori (ns-2, ns-3, OMNet i sl.) često daju različite rezultate prilikom testiranja protokola u odnosu na performanse realnih uređaja (laptop, android, čvorovi), zato bi ovakva vrsta emulatora približila i omogućila studentima i nastavnicima dalja istraživanja performansi WMN protokola. Ovakav koncept doprineo bi lakšem razumevanju WMN protokola u odnosu na primenu različitih algoritama za prenos ruting informacija između čvorova. Svaki čvor u emulatoru bio bi konfigurabilan i podržavao bi različite vrste WMN protokola.

Kreiranje ovakve vrste emulatora se zasniva na korišćenju virtualizacije na OS nivou, takozvana Linux kontejner virtualizacija (Scheepers, M. J., 2014).

Simulacija u ovom slučaju ne može realno da prikaže rezultate, pre svega zbog implementacije na primer OLSR protokola za ns-2. Druga stvar je da se rezultati u aplikacijama i u samom kernel razlikuju, što automatski pravi razliku između simulacije i realnog uređaja.

6.2.1. Pregled postojećih emulatora

Postoji nekoliko istraživanja na temu emulacije WMN, odnosno ad hok mreža.

U radu *VirtualMesh* (Staub et al., 2009). autor govori o razvojnom procesu za protokole u WMN mrežama, gde se ističe da je sam proces zavisan od evaluacija samog mrežnog simulatora i testiranju u realnom okruženju.

CORE (Common Open Research Emulator) emulator u realnom vremenu omogućava brzu primenu hibridnih tehnologija između realnog hardvera i virtuelnih mrežnih čvorova (Ahrenholz, J., et al., 2008). *CORE* koristi *FreeBSD* virtualizaciju i mrežni stek da bi omogućio izgradnju, planiranje i testiranje fizičkih mreža, bez potrebe za skupim hardverom.

U radu *CORE:A real time network emulator* (Ahrenholz, J., et al., 2008) prikazani su rezultati testiranja *CORE* emulatora i realnih fizičkih žičanih i bežičnih mreža. Pomoću

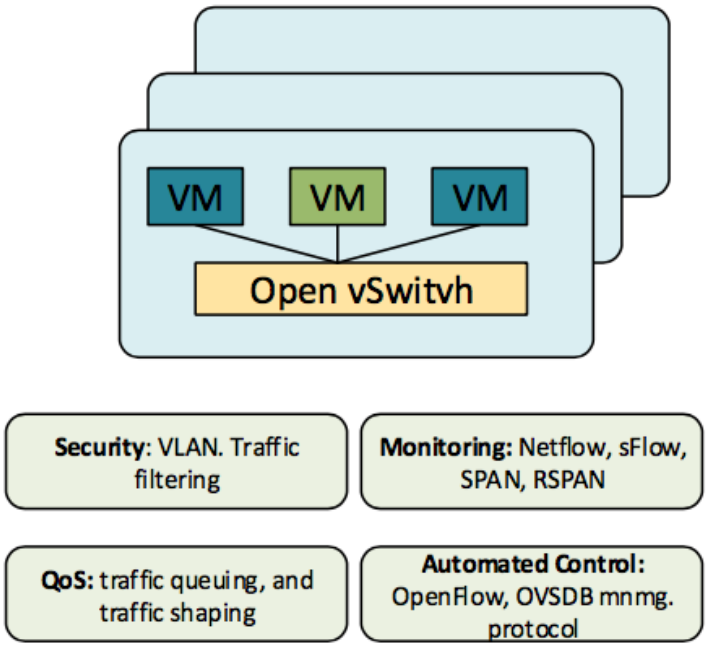
CORE emulatora na jednom serveru moguće je primiti i poslati više od 300.000 paketa u sekundi.

6.2.2. Emulatori mrežnog sloja u virtualnim okruženjima

Prelaskom na virtualizaciju stvoren je novi pristup na mrežnom sloju koji omogućava međusobno povezivanje virtualnih mašina (VM). VM mreže su nametnule zahteve u umrežavanju koji tradicionalno nisu raspoloživi. Oni pružaju prednosti na mrežnom sloju koji nisu prisutni na fizičkim mrežama. Mreže u virtuelnim sistemima u velikom broju su izgrađene na osnovu Eternet tehnologije, ali ova tehnologija ne zadovoljava potpuno zahteve virtualnih mreža (Pfaff et al., 2009).

Open vSwitch je softver koji emulira L2/L3 svič i namenjen je virtualnim mrežama (slika 6.1). Kompatabilan je s većinom Linux baziranih virtualnih okruženja uključujući *Xen, XsenServer, KVM i QEMU Proxmox VE VirtualBox.*

Multilejer virtualni svič, dizajniran da omogući mrežnu automatizaciju kroz programske ekstenzije, koji podržavaju standarde za upravljanjem interfejsima i protokole (NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag, 802.1Q).

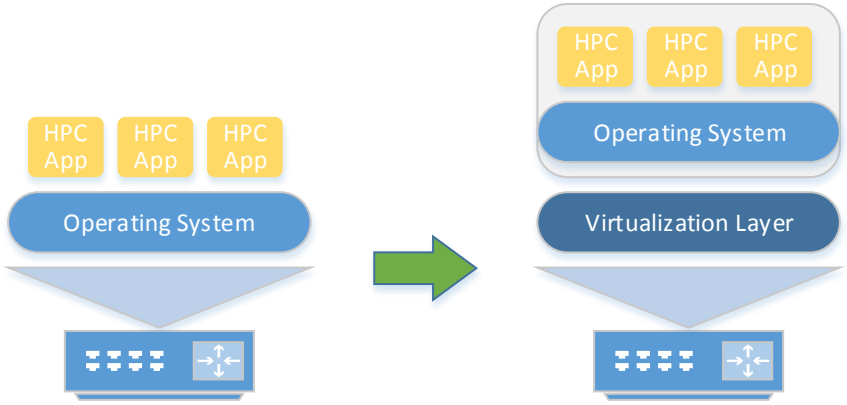


Slika 6.1 Open vSwitch

Squirrel emulator je programiran u *Go lang* programskom jeziku i emulira bežične mreže (IEEE 802.11) na Eternet infrastrukturi, što daje mogućnost da povežemo više linux hostova fizički ili virtualno i na kraju testiramo WMN protokole (802.11 ili ad hoc) (Vassis et al., 2005).

6.3. Upotreba virtualizacije za kreiranje WMN mreže

Pojam virtualizacija u računarstvu se odnosi na čin stvaranja virtualne verzije hardverske platforme, operativnog sistema, servera za skladištenje podataka, ili mrežnih resursa. Prednost virtualizacije je pre svega u uštedi resursa, ali i u smanjenju potrošnje električne energije. Primenom virtualizacije lakše se distribuiraju servisi i aplikacije u mreži, administracija je olakšana, i samam tehnologija pruža bezbednost na visokom nivou i otpornost na otkaze. Takođe, virtualizacija doprinosi očuvanju životne sredine, i spada u takozvane "zelene tehnologije".



Slika 6.2 Virtualizacija

Virtrualizacija je softver koji odvaja fizičke infrastrukture za stvaranje različitih namenskih servisa. To je osnovna tehnologija na kojoj se zasniva računarstvo u oblaku (Cloud computing). Virtualizacija omogućava pokretanje više operativnih sistema ili aplikacija na istom serveru u isto vreme. Ovaj koncept omogućava kompanijama da smanje troškove uz povećanje efikasnosti, korišćenje fleksibilnosti postojećeg računarskog hardvera. Osnovna razlika između virtualizacije i računarstva u oblaku je to da je virtualizacija softver za manipulaciju hardvera a računarstvo u oblaku proizilazi iz te manipulacije.

Generalno, postoji nekoliko osnovnih tipova:

- hardverska virtualizacija
- virtualizacija desktopa
- softverska virtualizacija
- memorijska virtualizacija
- virtualizacija podataka

- mrežna virtualizacija
- virtualizacija skladištenja podataka

6.3.1. Linux kontejneri

Server konsolidacija se odnosi na upotrebu fizičkog servera za podršku više servera ili korisničkih aplikacija. Server konsolidacija omogućava deljenje računarskih resursa između više aplikacija ili usluga istovremeno. Uglavnom se koristi za smanjenje broja potrebnih servera u organizaciji. Primarni cilj server konsolidacije je konzumiranje raspoloživih resursa servera u cilju smanjivanja kapitalnih i operativnih troškova. Statistički podaci govore da se iskorišćava samo 15% do 30% od ukupnog kapaciteta servera. Server konsolidacijom, stopa iskorišćenosti resursa servera može se podići na 80%. Virtualizacija nam omogućava server konsolidaciju, gde praktično imamo više virtuelnih server na jednom fizičkom serveru (Scheepers, M. J., 2014).

Virtualizacija na OS nivou je takozvana kontejner virtualizacija (Wes Felter et al., 2014). Većina aplikacija koje rade na serveru mogu da dele mašinu, a prednost je izolacija i sigurnost. U većini situacija različiti operativni sistemi nisu potrebni na istom serveru, već samo više instanci istog. Takozvana *OS-level* virtualizacija je dizajnirana da obezbedi potrebnu izolaciju i sigurnost za pokretanje više aplikacija ili više kopija OS-a na istom serveru.

Tehnologija softverskih kontejnera nije nova. FreeBSD je sistemom Jails ovu ideju prvi implementirao još 2000. godine. Oracle je nešto slično razvio u Solaris OS-u (Zones), dok se u skorije vreme u open source zajednici radilo na projektima OpenVZ i LXC (Linux Containers). Manje je poznato da Google ima i sopstvenu kontejnersku tehnologiju lmctfy (Let Me Contain That For You) koju koristi da korisnicima isporuči aplikacije kao što su Search, Gmail, Docs.

U razvojnom ciklusu neke aplikacije postoje mnoge prepreke. Tokom razvoja je glavni problem prilagođavanje razvojnog okruženja, gde je zavisnost od platforme ili neke tehnologije skoro neizbežna. *Docker* kao centralizovana servis orijentisana platforma pokušava da reši neke od ovih problema. Aplikacije se mogu razdeliti na upravljive, funkcionalne komponente, pojedinačno pakovane, koje se lako raspoređuju na različitim arhitekturama.

Kontejnerizacija nije novi koncept u računarskom svetu. Neki *Unix* bazirani sistemi već više od decenije koriste ovaj način za raspodelu resursa servera. *Linux* od 2008. poseduje podršku za kontejnerizaciju (*LXC – Linux Container*) u svom jezgru. *LXC* koristi *kernel cgroups* (omogućava izolaciju i praćenje pri raspodeli resursa), i *namespaces* (omogućava grupama da budu odvojene kako ne bi mogle "da vide" jedna drugu) u sprovođenju procesa izolacije.

Cilj kontejnera je potpuna standardizacija. To znači da kontejneri povezani na host koriste posebno definisane interfejse za svaki zadatak. Kontejnerska aplikacija ne bi trebalo da brine o detaljima hosta vezano za resurse ili arhitekturu. Ovo pojednostavljuje razvoj radnog okruženja. Isto tako, za host važi da je svaki kontejner posebna crna kutija. Host ne mari o detaljima aplikacije unutar kontejnera.

6.3.2. Docker

Docker je tehnologija za kreiranje softverskih kontejnera, kao što su paketi pojedinačnih aplikacija koji sadrže sve neophodno za pokretanje i izvršavanje. Na jednom serveru može da se izvršava više kontejnera istovremeno, ali sve mora da pokreće isti operativni sistem. U odnosu na upotrebu virtuelnih mašina, postavka sistema je manje kompleksna, na serveru je potrebno da postoji osnovni operativni sistem i softver koji omogućava upotrebu kontejnera, odnosno u konkretnom slučaju Docker Engine.

Docker je otvorena platforma za programere i sistem administratore za izgradnju, hostovanje i pokretanje distribuiranih aplikacija (James T., 2014). Sastoji se od *DockerEngine* i *DockerHub-a* klaud (Cloud) servisa za deljenje aplikacija i automatizaciju tokova. *Docker* omogućava aplikacijama da se brzo uklapaju s komponentama i eliminiše frakciju između razvojnog i proizvodnog procesa. Kao rezultat *Docker* može brže da isporuči i podigne aplikacije, nepromenjenim, na laptopovima, data centrima (VM) i na svakom klaudu.

S druge strane, za korišćenje tradicionalne virtualizacije potreban je odgovarajući hajpervizor, a svaka pojedinačna virtuelna mašina mora da ima instaliran sopstveni operativni sistem na kome se željena aplikacija izvršava. VM su zato fleksibilnije jer ne zavise od osnovnog operativnog sistema, ali su softverski kontejneri znatno manje kompleksni i efikasniji u iskorišćavanju resursa, što omogućava da se na istom serveru pokreće četiri do šest puta više kontejnera! To za kompanije znači ozbiljnu uštedu; u slučaju velikih kompanija, radi se o milionima dolara na godišnjem nivou.

Docker je učestvovao u razvoju LXC tehnologije i iskoristio je kao osnovu svog rešenja, a ključna razlika koju donosi na tržište jesu pouzdanost i laka upotreba. Deo toga je zahvaljujući bliskoj saradnji sa kompanijama Google, Parallels, Canonical, Red Hat i drugima, na standardizaciji osnovne open source komponente libcontainer.

Docker omogućava lako pokretanje kontejnera u klaud okruženju, odnosno napravio ih je tako da se lako integrišu u većinu DevOps aplikacija, uključujući Puppet, Chef, Vagrant i Ansible, i olakšava mnoge poslove koje obično obavljaju te aplikacije. Zato je Docker posebno koristan, na primer, da se pokrene više development okruženja s različitim podešavanjima, ili za sprovođenje testiranja, ili da se omogući rad više ljudi na istom projektu s identičnim podešavanjem.

Ukratko, kontejnerska tehnologija olakšava pokretanje više aplikacija na jednom serveru, što znači ozbiljne uštede za korisnike, kao i jednostavnije održavanje sistema. Docker je najbolja implementacija ove tehnologije, koja je početkom februara stigla do verzije 1.5 i kao najbitniju novost donela podršku za IPv6. Očekuje se njen dalji ubrzan razvoj i sve šira implementacija, posebno u data centrima velikih kompanija.

6.4. Emulator MeshLAB

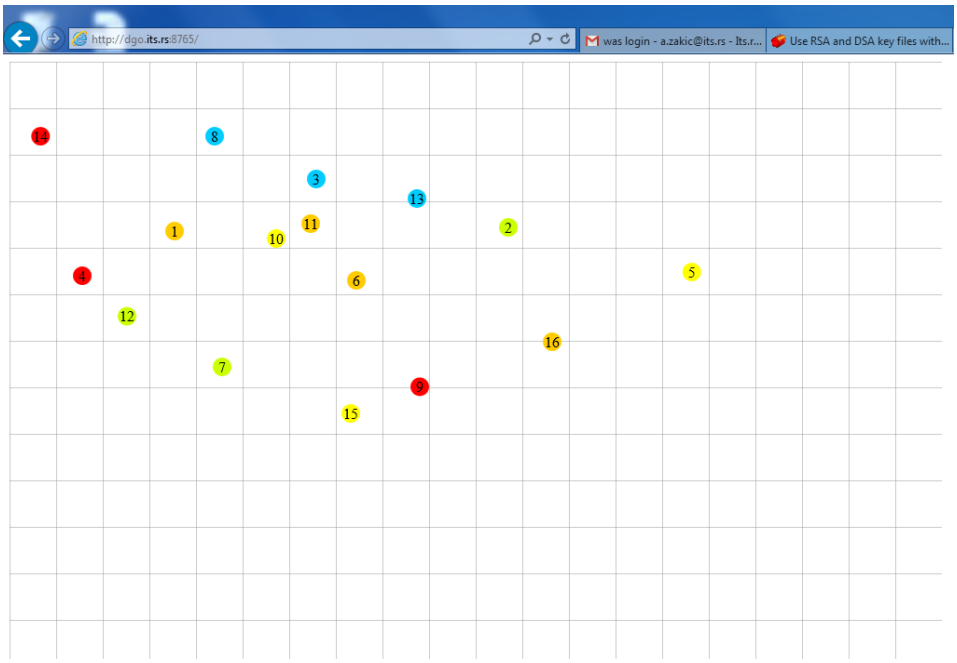
MeshLAB softver se zasniva na virtualizaciji na OS nivou i emulatora za bežične mreže. Za implementaciju OS level virtualizacije korišćen je računarski oblak (Cloud). Najveći globalni isporučioci usluga računarskog oblaka su Amazon (AWS), IBM, Google, Microsoft.

Računarski oblak je model koji omogućava svuda prisutan, pogodan mrežni pristup deljivim računarskim resursima (mrežnim, serverima, skladištu podataka, aplikacijama i servisima), koji na zahtev korisnika i uz minimalnu interakciju s isporučiocem usluga mogu biti brzo stavljeni na raspolaganje korisniku ili otkazani (Mell et al., 2009).

MeshLAB (slika 6.5) poseduje web interfejs koji daje vizuelni pregled WMN čvorova: levim tasterom miša pravi se proizvoljni raspored WMN čvorova u mreži. Sam emulator na mrežnom sloju je sposoban da u zavisnosti od rastojanja čvorova simulira i nivo jačine signala između čvorova, pa na osnovu toga MeshLAB softver omogućava kreiranje proizvoljne mreže bežičnih čvorova.

Ako je svaki čvor Linux kontejner povezan s drugima pomoću nekog softverskog switch-a (Open vSwitch ili sl.) preko virtuelnih interfejsa (tun/tap) kontejnera, dobijamo mesh mrežu Linux kontejnera.

Za kreiranje mreže kontejnera koristi se Docker platforma. Pomoću Docker-a se pokreću, izvršavaju ili stopiraju procesi unutar kontejnera.



Slika 6.3 Linux kontejneri – MeshLAB

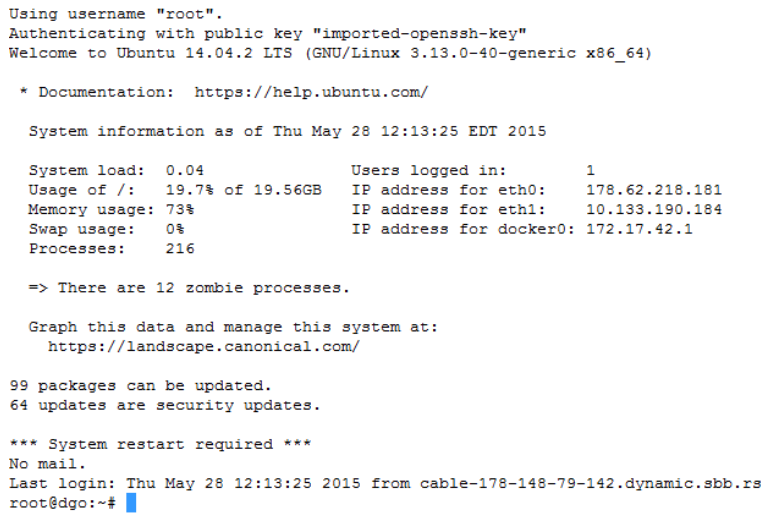
MEshLAB kao softver omogućava instalaciju i konfiguraciju WMN protokola (B.A.T.M.A.N, OLSR, Babel, BmX6 i dr.).

6.5. Rešenje

Za implementaciju MeshLAB okruženja korišćen je Ubuntu Linux, a moguć je i na drugim Linux distribucijama (Fedora, CentOS, Slackware, Debian). MeshLAB je osposobljen da radi i na virtuelnim mašinama s hardverskom virtualizacijom (Xen, KVM i sl.).

Da bi aplikacija bila dostupna onlajn, u ovom slučaju korišćena je Cloud infrastruktura Digital Ocean hosting kompanije.

Instaliran je Ubuntu 14.04 server sa KVM hardverskom virtualizacijom. Karakteristike virtuelne mašine su: 512MB RAM, 1 Core CPU, 20GB SSD, što zadovoljava kreiranje više od 90 čvorova (Slika 6.6).



Slika 6.4 Virtual Private Server

Instalacija Docker platforme za Ubutnu 14.04:

```
$ which wget

$ sudo apt-get update $ sudo apt-get install wget

$ wget -qO- https://get.docker.com/ | sh
```

Pokretanje MeshLAB okruženja:

Nakon instalacije Docker-a, potrebno je da se napravi Dockerfile kofiguracija. U ovom slučaju, ona izgleda ovako:

```
$ ################################################################

# Dockerfile to build MeshLab container images

# Based on Ubuntu

################################################################

# Set the base image to Ubuntu

FROM ubuntu

# File Author / Maintainer

MAINTAINER Aleksandar Z.

#ADD run  /usr/local/sbin/run

#RUN chmod 755 /usr/local/sbin/run
```

```
VOLUME /root/meshlab

# Update the repository sources list

RUN apt-get update

RUN DEBIAN_FRONTEND=noninteractive apt-get -y install python

# Install wget iperf

RUN apt-get install -y wget iperf supervisor nano olsrd olsrd-plugins

RUN cd /usr/local/bin && wget 172.17.42.1:9999/squirrel-worker && chmod +x
squirrel-worker

#WORKDIR

# Set the default command to execute

#CMD

#COPY . /root/meshlab

#CMD ["python3", "/root/meshlab/run1.py"]

#RUN script

#CMD
```

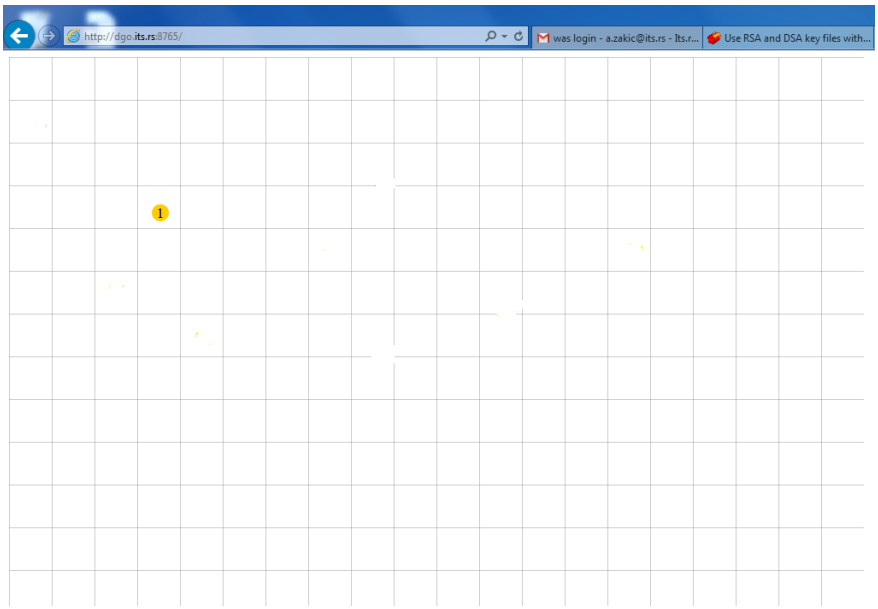
Pravljenje slike (Build image) u Dockeru:

```
$ sudo docker build -t meshlabnode .
```

Posle kreiranje slike kontejnera na lokalnom disku, potrebno je pokrenuti instancu:

```
$ sudo docker run -i --privileged --name node1 -i -t -d meshnode001
```

Ovo je primer kreiranja jednog kontejnera, odnosno WMN čvora (Slika 6.7).



Slika 6.5 WMN čvor

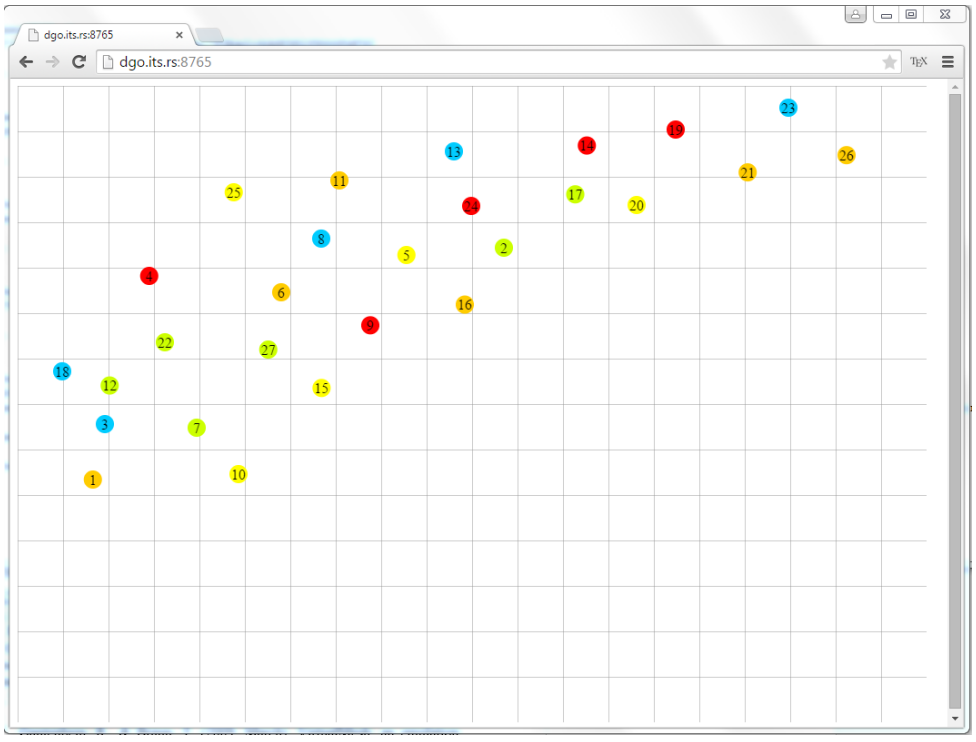
Izvršna skripta u programskom jeziku Python pomoću koje se automatizuje proces generisanja rednog broja čvora:

```
file = open('test', 'r') #otvara fajl "test" u read modu

x = file.read() #promenjivoj "x" dodeljujemo String iz fajla

z = int(x)#konvertuje String u Intiger

#print z #Ispisuje vrednost

z = z+1 #Svaki put kada se skripta izvrsi povecava vrednost

#za jedan tako da se dva broja nikad ne ponavljaju

#Samo ovde ubacite svoj kod u kome dodeljujete

#vrednost kontejnerima

file = open('test', 'w')#otvara fajl "test" u write modu

x = str(z)#konvertuje promenjivu "z" u string

file.write(x)#upisuje vrednost u fajl

#print x#ispisuje vrednost za "x" -
```

Izvršna skripta koja pravi instance čvora i pokreće *tap* interfejs pomoću koga se čvor povezuje na mesh mrežu.

```
#!/bin/bash

eval "$(python /root/meshlab/main.py)"

file="/root/meshlab/test" #the file where you keep your string name

name=$(cat "$file")        #the output of 'cat $file' is assigned to the $n
ame variable

docker run -i --privileged --name node$name -i -t -d meshnode001

#docker start node$name

docker exec -it node$name squirrel-worker -m 172.17.42.1:1234 -i $name -t t
ap0

docker exec -id node$name olsrd -f /olsrdmeshlab.conf
```

Kao rezultat stvorena je WMN mrežu čvorova (Slika 6.8). U zavisnosti od hardvera i količine RAM memorije, moguće je kreirati i više od 1.000 čvorova.



Slika 6.6 WMN mreža čvorova

```
root@dgo:~# docker ps

CONTAINER ID        IMAGE                       COMMAND             CREATED
STATUS              PORTS                     NAMES

8b330173b6db        meshnode001:latest         "/bin/bash"         11 minut
es ago      Up 11 minutes                             node28

8b6f19a230d3        meshnode001:latest         "/bin/bash"         11 minut
es ago      Up 11 minutes                             node27

af1e76edb4b2        meshnode001:latest         "/bin/bash"         11 minut
es ago      Up 11 minutes                             node26

7a6c5725cfb9        meshnode001:latest         "/bin/bash"         11 minut
es ago      Up 11 minutes                             node25

60c2280f50b5        meshnode001:latest         "/bin/bash"         11 minut
es ago      Up 11 minutes                             node24

2bddb33bbd08        meshnode001:latest         "/bin/bash"         12 minut
es ago      Up 12 minutes                             node23

b1b4873de8f6        meshnode001:latest         "/bin/bash"         12 minut
es ago      Up 12 minutes                             node22

33227651d235        meshnode001:latest         "/bin/bash"         13 minut
es ago      Up 13 minutes                             node21

1c4ba411f53d        meshnode001:latest         "/bin/bash"         13 minut
es ago      Up 13 minutes                             node20

60b16c3e4272        meshnode001:latest         "/bin/bash"         13 minut
es ago      Up 13 minutes                             node19

8eb3ed086480        meshnode001:latest         "/bin/bash"         13 minut
es ago      Up 13 minutes                             node18

5583d277b949        meshnode001:latest         "/bin/bash"         14 minut
es ago      Up 14 minutes                             node17

1b6dc8a657d2        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                 node16

35aabda8156e        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                 node15

a38ae6bb997a        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                 node14

6d0f1fb0a6c2        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                 node13
```

```
cb26e540f004        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node12

ce0cc1409520        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node11

153c698fab4f        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node10

591dfaed66db        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node9

a15e1f10c768        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node8

5f990f027f25        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node7

699a2ab5df5c        meshnode001:latest         "/bin/bash"         8 days a
go          Up 8 days                                  node6

2dd86ceefee7        ubuntu:14.04               "/bin/bash"         3 months
ago         Up 3 months                                node5

a8435e19f7d1        ubuntu:14.04               "/bin/bash"         3 months
ago         Up 3 months                                node4

7d01a9034795        ubuntu:14.04               "/bin/bash"         3 months
ago         Up 3 months                                node3

1a7603a80d0c        ubuntu:14.04               "/bin/bash"         3 months
ago         Up 3 months                                node2

5c702cbf9fb4        ubuntu:14.04               "/bin/bash"         3 months
ago         Up 3 months                                node1
```

6.6. Analiza

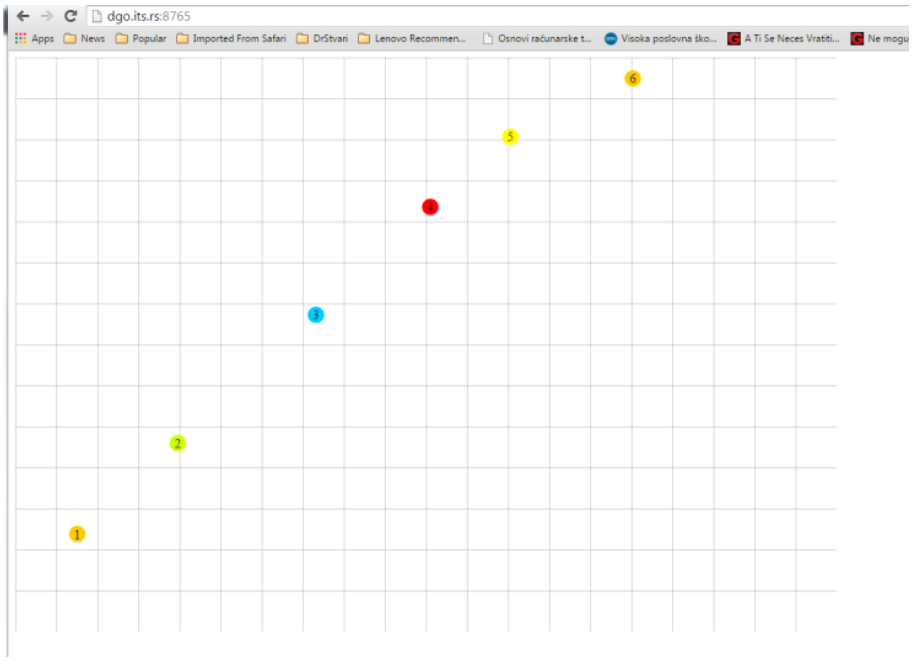
Nakon instalacije MeshLAB okruženja sledi analiza i pregled laboratorijskih vežbi. Iz prethodnog se videlo da testiranje WMN protokola nije jednostavan proces. Postoje komjuniti mreže koje su velike i na kojima se mogu primeniti neki eksperimenti (BMx6 na primer). MeshLAB okruženje je nakon brojnih simulatora jedino rešenje koje pre svega može da se koristi za upoređivanje i analizu WMN protokola. Prednost je što je emulator sposoban da podrži i nekoliko hiljada mobilnih ili statičkih čvorova.

6.7. Kreiranje jednostavne WMN mreže pomoću MeshLAB

U prethodnom poglavlju se jasno vidi koja je prednost korišćenja ovakvih tipova laboratorija. Multiplaformsko okruženje u oblaku (Cloud) je pre svega centralizovano

(na jednom mestu), moguće je promeniti ili dodati hardverske resurse, implementirati ga na nekim manjim (Raspberry Pi 2) ili većim sistemima (Dedicated serveri). Ovo omogućava prilagođavanje okruženja u zavisnosti od potrebe (Zakić et al. 2015).

Kreiranje mreže u MeshLAB okruženju je jednostavno klikom na komandno dugme *Kreiraj čvor* (slika 6.9), prilikom čega se izvršava skripta *createNODE*.



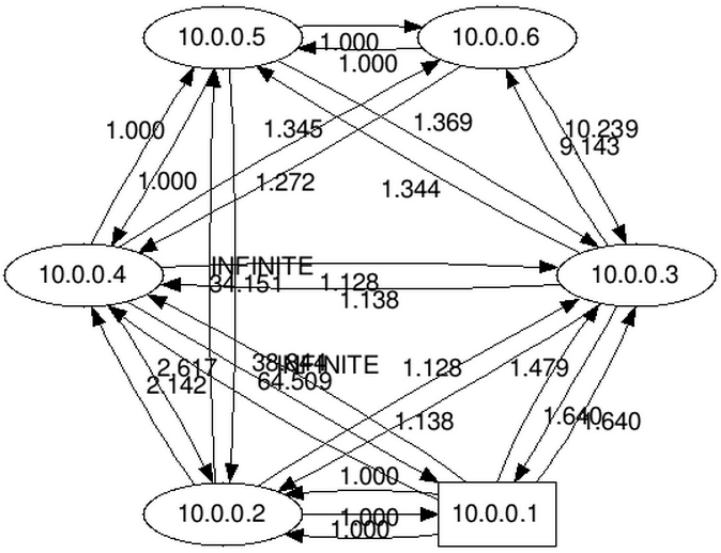
Slika 6.7 Kreiranje čvorova za WMN mrežu

Ovo je primer gde su čvorovi povezani redno, a protokol je OLSR. Da bi bilo jasno kako se vrednosti kvaliteta linka (LQ) menjaju, kreiraćemo grafikon (Slika 6.10). Kreiranje grafikona omogućava plugin *olsr_dot_draw*.

```
root@dgo:~# cat olsrdmeshlab.conf

Interface "tap0"

{

Ip4Broadcast 255.255.255.255

}

LoadPlugin "olsrd_dot_draw.so.0.3"

{

PlParam "accept" "127.0.0.1"

PlParam "port" "2004"
```

```
}

root@dgo:~# docker exec -i node1 telnet localhost 2004

Trying ::1...

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Connection closed by foreign host.

digraph topology

{

"10.0.0.1" -> "10.0.0.4"[label="64.509", style=solid];

"10.0.0.1" -> "10.0.0.3"[label="1.640", style=solid];

"10.0.0.1"[shape=box];

"10.0.0.1" -> "10.0.0.2"[label="1.000", style=solid];

"10.0.0.1"[shape=box];

"10.0.0.1" -> "10.0.0.2"[label="1.000"];

"10.0.0.1" -> "10.0.0.3"[label="1.640"];

"10.0.0.1" -> "10.0.0.4"[label="INFINITE"];

"10.0.0.2" -> "10.0.0.1"[label="1.000"];

"10.0.0.2" -> "10.0.0.3"[label="1.138"];

"10.0.0.2" -> "10.0.0.4"[label="2.142"];

"10.0.0.2" -> "10.0.0.5"[label="34.151"];

"10.0.0.3" -> "10.0.0.1"[label="1.479"];

"10.0.0.3" -> "10.0.0.2"[label="1.128"];

"10.0.0.3" -> "10.0.0.4"[label="1.138"];

"10.0.0.3" -> "10.0.0.5"[label="1.344"];

"10.0.0.3" -> "10.0.0.6"[label="9.143"];
```

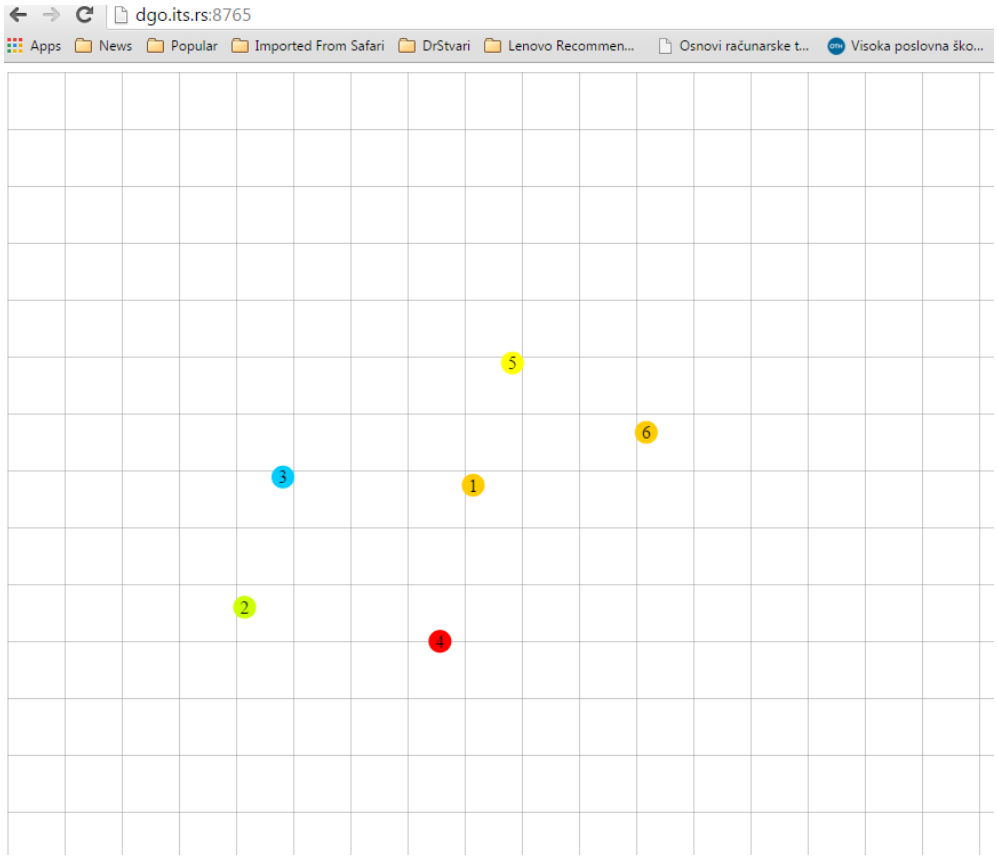
```
"10.0.0.4" -> "10.0.0.1"[label="38.844"];

"10.0.0.4" -> "10.0.0.2"[label="2.617"];

"10.0.0.4" -> "10.0.0.3"[label="1.128"];

"10.0.0.4" -> "10.0.0.5"[label="1.000"];

"10.0.0.4" -> "10.0.0.6"[label="1.272"];

"10.0.0.5" -> "10.0.0.2"[label="INFINITE"];

"10.0.0.5" -> "10.0.0.3"[label="1.369"];

"10.0.0.5" -> "10.0.0.4"[label="1.000"];

"10.0.0.5" -> "10.0.0.6"[label="1.000"];

"10.0.0.6" -> "10.0.0.3"[label="10.239"];

"10.0.0.6" -> "10.0.0.4"[label="1.345"];

"10.0.0.6" -> "10.0.0.5"[label="1.000"];

}
```



Slika 6.8 GraphViz šema za datu topologiju

Kao što se vidi na Slici 6.10, vizuelni prikaz se podudara s vrednostima koje su dobijene korišćenjem plugina *olsr_dot_draw*.

Ali, ako se promeni fizička topologija (Slika 6.11), rezultati su potpuno drugačiji.

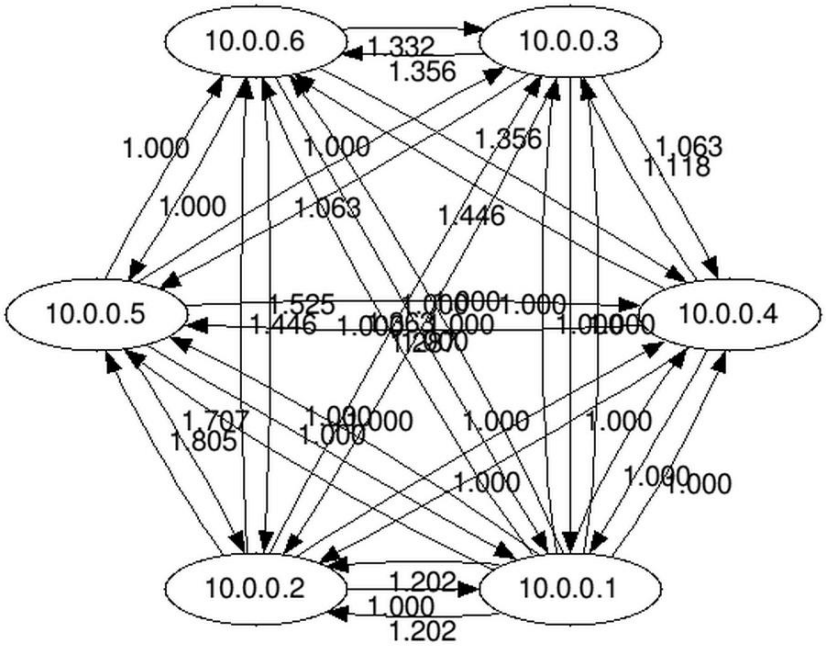


Slika 6.9 Promenjena fizička topologija WMN mreže

```
root@dgo:~# docker exec -i node1 telnet localhost 2004

Trying ::1...

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Connection closed by foreign host.

digraph topology

{

"10.0.0.1" -> "10.0.0.4"[label="1.000", style=solid];

"10.0.0.1" -> "10.0.0.5"[label="1.000", style=solid];

"10.0.0.1" -> "10.0.0.6"[label="1.000", style=solid];

"10.0.0.1" -> "10.0.0.3"[label="1.000", style=solid];
```

```
"10.0.0.1" -> "10.0.0.2"[label="1.202", style=solid];
"10.0.0.1" -> "10.0.0.2"[label="1.202"];
"10.0.0.1" -> "10.0.0.3"[label="1.000"];
"10.0.0.1" -> "10.0.0.4"[label="1.000"];
"10.0.0.1" -> "10.0.0.5"[label="1.000"];
"10.0.0.1" -> "10.0.0.6"[label="1.000"];
"10.0.0.2" -> "10.0.0.1"[label="1.000"];
"10.0.0.2" -> "10.0.0.3"[label="1.000"];
"10.0.0.2" -> "10.0.0.4"[label="1.000"];
"10.0.0.2" -> "10.0.0.5"[label="1.805"];
"10.0.0.2" -> "10.0.0.6"[label="1.446"];
"10.0.0.3" -> "10.0.0.1"[label="1.000"];
"10.0.0.3" -> "10.0.0.2"[label="1.000"];
"10.0.0.3" -> "10.0.0.4"[label="1.063"];
"10.0.0.3" -> "10.0.0.5"[label="1.000"];
"10.0.0.3" -> "10.0.0.6"[label="1.356"];
"10.0.0.4" -> "10.0.0.1"[label="1.000"];
"10.0.0.4" -> "10.0.0.2"[label="1.000"];
"10.0.0.4" -> "10.0.0.3"[label="1.118"];
"10.0.0.4" -> "10.0.0.5"[label="1.287"];
"10.0.0.4" -> "10.0.0.6"[label="1.446"];
"10.0.0.5" -> "10.0.0.1"[label="1.000"];
"10.0.0.5" -> "10.0.0.2"[label="1.707"];
"10.0.0.5" -> "10.0.0.3"[label="1.063"];
"10.0.0.5" -> "10.0.0.4"[label="1.363"];
"10.0.0.5" -> "10.0.0.6"[label="1.000"];
```

```
"10.0.0.6" -> "10.0.0.1"[label="1.000"];

"10.0.0.6" -> "10.0.0.2"[label="1.525"];

"10.0.0.6" -> "10.0.0.3"[label="1.332"];

"10.0.0.6" -> "10.0.0.4"[label="1.356"];

"10.0.0.6" -> "10.0.0.5"[label="1.000"];

}
```



Slika 6.10 GraphViz šema za datu topologiju

U ovom slučaju se dokazuje da MeshLAB okruženje ima karakteristike realne WMN mreže (Slika 6.12). Pomoću MeshLAB okruženja moguće je testirati i druge WMN protokole kao što su Babel, B.A.T.M.A.N., BMx6, a ti testovi biće prikazani u budućim istraživanjima.

7. PRIMENA SPEKTRALNE TEORIJE GRAFOVA U WMN MREŽAMA

Teorija grafova je relativno mlada oblast matematike. Najveći napredak dostignut je poslednjih decenija, zahvaljujući savremenoj računarskoj tehnologiji. Graf je matematička struktura koja se koristi pri modeliranju relacija između objekata nekog skupa.

Najstariji problem u teoriji grafova jeste problem Kenigzberških mostova. Naime, švajcarski matematičar Ojler (Leonhard Euler) je 1736. godine naišao na sledeći problem: grad Kenigzberg leži na obalama i na dva ostrva reke Pregel, koji su povezani sa sedam mostova. Pitanje je bilo da li je moguće obići sve mostove tačno jednom, polazeći iz bilo koje tačke.

U spektralnoj teoriji, grafovi se izučavaju koristeći sopstvene vrednosti matrice M, koja na neki način opisuje svaki graf. Tako se može govoriti o matrici susedstva A, Laplasovoj matrici L, matrici rastojanja D, neoznačenoj Laplasovoj matrici Q itd. (Cvetkovic et al. 1995). Spektri grafova i odgovarajući sopstveni vektori imaju značajne primene kod modeliranja i pretraživanja Interneta, obrada slika i prepoznavanja oblika, klasterizaciji podataka, u multiprocesorskim povezujućim mrežama, u socijalnim mrežama, u matematičkoj hemiji, ekonomiji i drugim naukama (Cvetkovic et al., 2009).

Jedan od najilustrativnijih primera je Google PageRank algoritam koji je zasnovan na iterativnom postupku nalaženja najveće sopstvene vrednosti matrice susedstva. Naime, stranice na Internetu se rangiraju na osnovu međusobnih linkova, a važnost neke stranice proporcionalna je odgovarajućoj komponenti Peronovog sopstvenog vektora. Analiziraju se koeficijenti Laplasovog karakterističnog polinoma i najveća sopstvena vrednost matrice rastojanja, kao i dve invarijante koje su bazirane na spektru grafova – energija i Estradin indeks.

Topološki indeksi i grafovske invarijante bazirane na sopstvenim vrednostima i rastojanjima između čvorova su veoma zastupljeni u računarstvu i matematičkoj hemiji. Njima se modeliraju razne osobine molekula i njihovih veza. Mnogi indeksi ostvaruju odličnu korelaciju između fizičkih, hemijskih, termodinamičkih i bioloških parametara hemijskih jedinjenja. Wienerov indeks jedna je od najstarijih i najpoznatijih grafovskih invarijanti. Definisan je 1947. godine kao zbir rastojanja između svih parova čvorova

$$W(G) = \sum_{u,v \in V} d(u,v)$$

gde d(u, v) predstavlja najkraće rastojanje između čvorova u i v (teorijski rezultati i primene prikazane su u Debrynin et al., 2001.

7.1. Formalne definicije

Graf G je uređen par (V, E). elementi skupa V se zovu čvorovi, a elementi skupa E grane grafa G.

Težinski graf je uređena trojka (V, E, w), gde je V skup čvorova, E skup grana, a w je funkcija težine. Težina grane e, koja povezuje čvorove u i v, označava se sa w(u, v) ili wu,v. Ako je graf neorijentisan, onda važi wu,v = w v,u. Ako je graf bez petlji, onda važi w u,u = 0.

Za dati graf G = (V, E), za bilo koja dva čvora $u, v \in V$, putanja od u do v je sekvenca čvorova (v_0, v_1,.., v_k), tako da je v_0 = u, v_k = v, i (v_i, v_{i+1}) je jedna grana iz skupa E, za svako $0 \leq i \leq k-1$, Ceo broj k je dužina putanje. Putanja je zatvorena ako je u = v. Graf G je jako povezan ako za svaka dva čvora (u, v) postoji putanja od u do v.

Red grafa G određen je brojem čvorova u grafu. Stepen čvora k jednak je broju grana koje su povezane s datim čvorom. Graf G je regularan stepena r ako svaki čvor ima stepen r. Stepen čvora v_i, označava se sa $d(v_i)$, definiše se kao broj suseda čvora v_i.

Graf G je regularan ako svi čvorovi grafa imaju isti stepen.

7.2. Spektralna teorema

Spektralna teorema grafa zasniva se na sopstvenim vrednostima i sopstvenim vektorima grafa.

Sopstvena vrednost (eigenvalue) matrice A je realan broj λ, ako matrična jednačina:

$$Ax = \lambda x$$

ima netrivijalno rešenje, koje nazivamo sopstveni vektor (eigenvector). Sopstvene vrednosti grafa su sopstvene vrednosti matrice susedstva.

Spektar grafa G određen je skupom sopstvenih vrednosti matrice susedstva za dati graf.

Spektralna teorija grafa koristi spektre nekih specifičnih matrica, kao što su matrica susedstva, Laplasova matrica i normalizovana Laplasova matrica, da bi se dobile određene informacije o grafu. Za određene porodice grafova moguće je izvršiti karakterizaciju grafa pomoću spektra grafa.

Energija grafova definisana je kao suma apsolutnih vrednosti sopstvenih vrednosti matrice susedstva (Zakić et al., 2014).

$$E(G) = \sum_{i=1}^{n} |\lambda_1|$$

7.3. Matrica susedstva i Laplasova matrica

Za besežinski, neusmeren graf G, matrica susedstva se definiše na sledeći način:

$$A_{i,j} = \begin{cases} 1, ako\ postoji\ grana\ od\ i\ do\ j \\ 0, ostalo \end{cases}$$

Normalizovana matrica susedstva je određena sa $\hat{A} = \sqrt{D^{-1}}\, A \sqrt{D^{-1}}$

Za dati usmereni graf G = (V, E), incidentna matrica B(G) je matrica dimenzije m x n, čiji su elementi b_{ij} određeni sa:

$$b_{ij} = \begin{cases} +1, ako\ je\ s(e_j) = v_i \\ -1, \quad ako\ je\ t(e_j) = v_i \\ 0, ostalo \end{cases}$$

Za dati neusmereni graf G = (V, E), incidentna matrica B(G) je matrica dimenzije m x n, čiji su elementi b_{ij} određeni sa:

$$b_{ij} = \begin{cases} +1, ako\ je\ v_i\ povezan\ granom\ e_j \\ \\ 0, ostalo \end{cases}$$

Matrica stepena D je dijagonalna matrica s vrednostima stepena čvorova $d_{i=1,2\ldots n}$ na dijagonali.

Laplasova matrica je definisana sa:

$$L = D - A$$

Sopstvene vrednosti matrice L nazivaju se Laplasove sopstvene vrednosti i uređene su u nerastućem poretku:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n-1} \geq \lambda_n = 0.$$

Ako je G usmeren graf i ako je B incidentna matrica od G, A matrica susedstva, a D dijagonalna matrica stepena, tada je(Gallier, J., 2014):

$$BB^T = D - A$$

Ako je graf G neusmeren, onda važi:

$$BB^T = D + A$$

Očigledno je da je suma elemenata u svakom redu matrice L jednaka nuli.

Normalizovana Laplasova matrica je određena sa:

$$L_N = D^{-1/2} A D^{-1/2}$$

Primer:

Ako je data matrica susedstva:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Sopstvene vrednosti matrice susedstva su $(-2, 1-\sqrt{5}, 0, 0, 1+\sqrt{5})$.

Normalizovana matrica susedstva je:

$$\widehat{A} = \begin{bmatrix} 0 & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} \\ \frac{1}{2\sqrt{3}} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2\sqrt{3}} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2\sqrt{3}} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2\sqrt{3}} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \end{bmatrix}$$

Sopstvene vrednosti normalizovane matrice susedstva su $(-\frac{2}{3}, -\frac{1}{3}, 0, 0, 1)$.

Laplasova matrica je:

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & -1 & 0 & -1 & 3 \end{bmatrix}$$

Sopstvene vrednosti Laplasove matrice su $(0, 3, 3, 5, 5)$.

Normalizovana Laplasova matrica je:

$$\hat{L} = \begin{bmatrix} 1 & \frac{-1}{2\sqrt{3}} & \frac{-1}{2\sqrt{3}} & \frac{-1}{2\sqrt{3}} & \frac{-1}{2\sqrt{3}} \\ \frac{-1}{2\sqrt{3}} & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ \frac{-1}{2\sqrt{3}} & \frac{1}{3} & 1 & \frac{1}{3} & 0 \\ \frac{-1}{2\sqrt{3}} & 0 & -\frac{1}{3} & 1 & \frac{1}{3} \\ \frac{-1}{2\sqrt{3}} & -\frac{1}{3} & 0 & -\frac{1}{3} & 1 \end{bmatrix}$$

Sopstvene vrednosti normalizovane Laplasove matrice su $(0,\ 1, 1, \frac{4}{3}, \frac{5}{3})$.

7.4. Spektralna klasterizacija

Specifične topološke osobine grafa koriste se za karakterisanje povezanosti i imaju značajan uticaj na dinamičke procese u kompleksnim mrežama, pa se analiza i sinteza ovih mreža zasniva na upotrebi metrike izražene relevantnim topološkim karakteristikama.

Osnovne strukturne osobine grafa mogu se proučavati razmatranjem topologije grafa. Topologija grafa određuje kako su čvorovi grafa međusobno povezani i u kojim su međusobnim relacijama.

Svaki čvor u grafu može se okarakterisati određenim osobinama, kao što je na primer vreme procesiranja, a svaka grana u grafu se može specificirati kao skup težinskih funkcija, kao na primer kašnjenje, propusni opseg, gubitak paketa...

Metrika je topološka, ako je moguće izračunati samo pomoću matrice susedstva. Topološka metrika može se klasifikovati na metrike koje su zasnovane na

- udaljenosti
- povezanosti
- spektaru grafa

U nastavku je prikazano kako se spektar grafa koristi u procesu klasterizacije složenih mreža.

Zbog svojih osobina spektar Laplasove matrice ima veću važnost za većinu fizičkih i hemijskih procesa, nego spektar matrice susedstva (B. Mohar et al. 1991).

Ako graf G nije povezan, onda broj sopstvenih vrednosti Laplasove matrice, koji su jednaki nuli, određuje broj komponenti grafa G. Za svaku matricu grafa G, spektar je jednak uniji spektra svake komponente grafa.

Druga najmanja sopstvena vrednost Laplasove matrice grafa naziva se algebarska povezanost grafa G i dobar je parametar za merenje povezanosti grafa. Na primer, druga najmanja vrednost je pozitivna samo ako je graf povezan (M. Fiedler, 1973). Vektor sopstvenih vrednosti Laplasove matrice koji odgovara drugoj najmanjoj sopstvenoj vrednosti naziva se Fiedler-ov vektor.

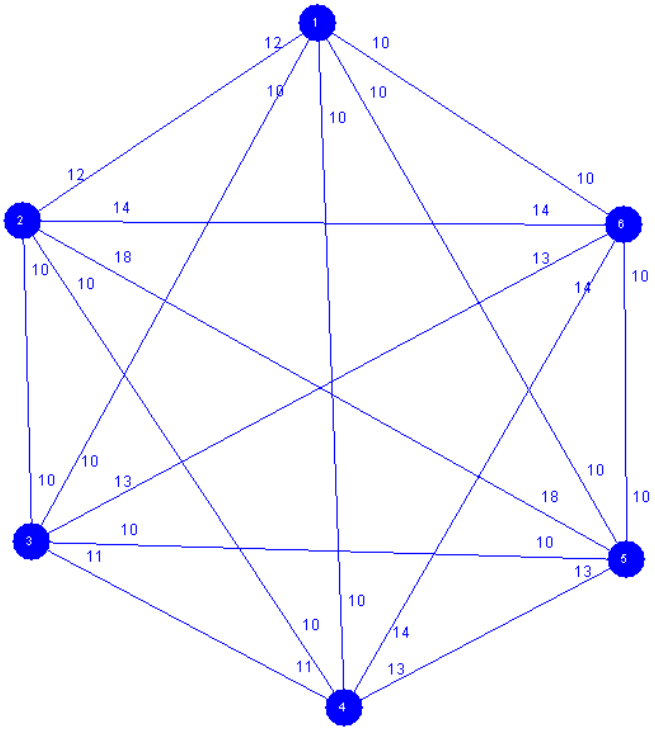
Algoritam spektralne klasterizacije
1. *Konstruiše se graf* 2. *Odredi se matrica susedstva* 3. *Odredi se Laplasova matrica i izračunaju se sopstvene vrednosti i sopstveni vektori* 4. *Posmatra se sopstveni vektor koji odgovara drugoj najmanjoj sopstvenoj vrednosti*

5. *Čvorovi grafa koji odgovaraju pozitivnim vrednostima posmatranog sopstvenog vektora pridružuju se jednom klasteru, a čvorovi koji odgovaraju negativnim vrednostima pridružuju se drugom klasteru.*
6. *Ako je potrebno podeliti graf na više klastera, onda se koraci 2-5 ponavljaju rekurzivno.*

7.5. Klasterizacija u MeshLAB eksperimentu

U nastavku je izvršena spektralna klasterizacija mesh mreže iz primera sa Slike 6.11.

Data mesh mreža može da se predstavi povezanim neusmerenim težinskim grafom kao na Slici 7.1. Mesh čvorovi predstavljeni su čvorovima grafa, a vrednosti OLSR metrike, koja predstavlja jačinu signala između WMN čvorova, predstavljene su granama grafa (Slika 7.1).

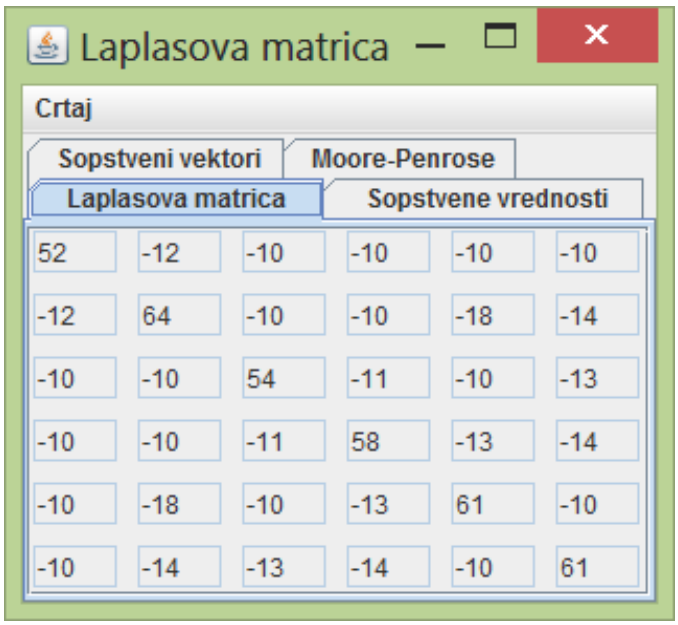


Slika 7.1 WMN mreža predstavljena neusmerenim težinskim grafom

Matrica susedstva datog grafa prikazana je na slici 7.2

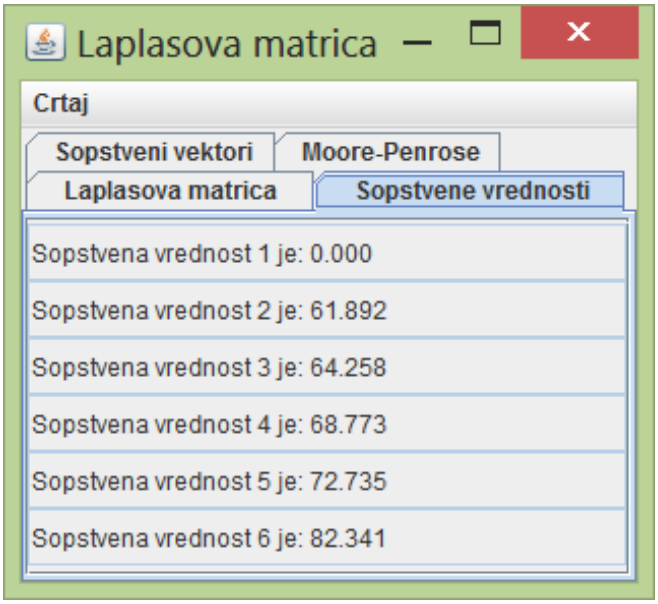
Slika 7.2Matrica susedstva

Laplasova matrica za graf sa slike je prikazana na slici 7.3:



Slika 7.3 Laplasova matrica

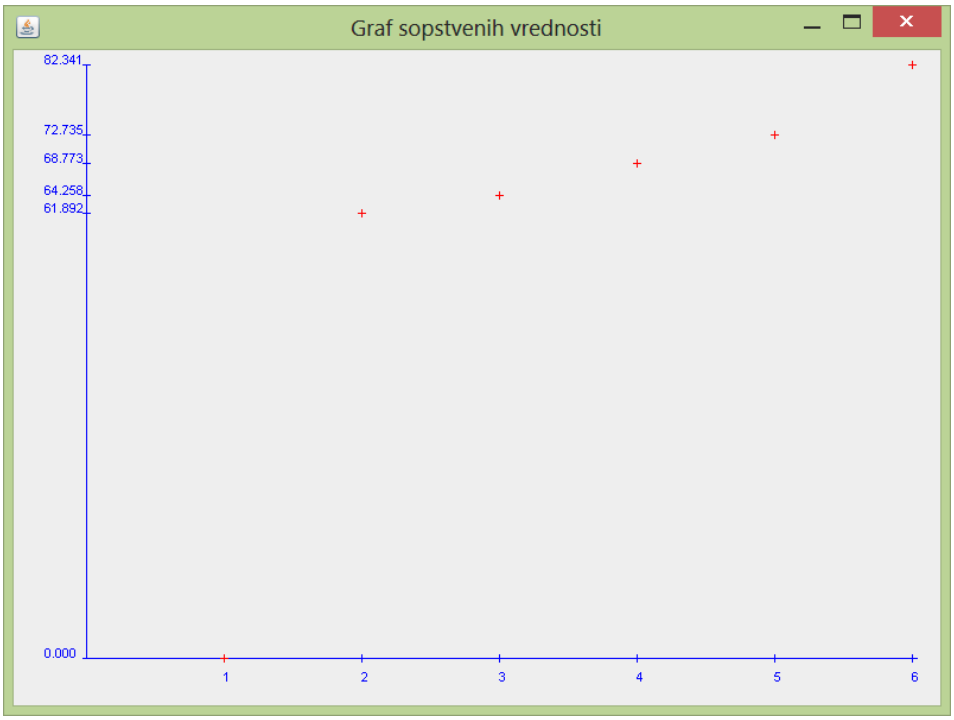
Sopstvene vrednosti Laplasove matrice prikazane su na slici 7.4:



Slika 7.4Sopstvene vrednosti Laplasove matrice

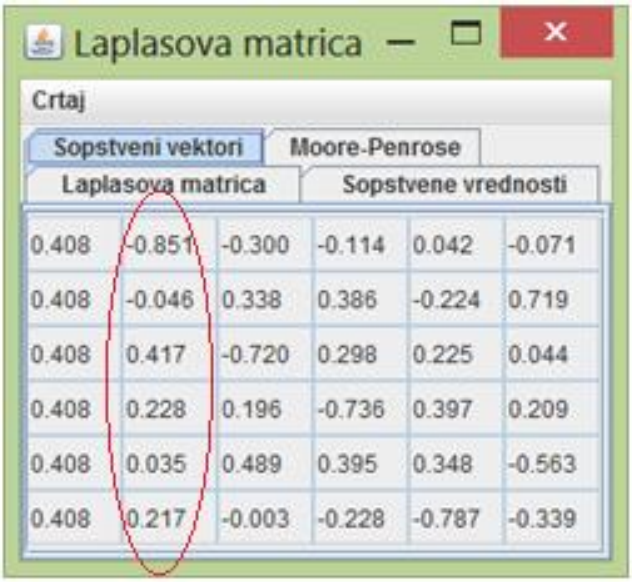
Vrednost nula prve sopstvene vrednosti ukazuje da WMN čvorovi u mesh mreži potpuno povezani.

Graf sopstvenih vrednosti Laplasove matrice prikazan je na slici 7.5:



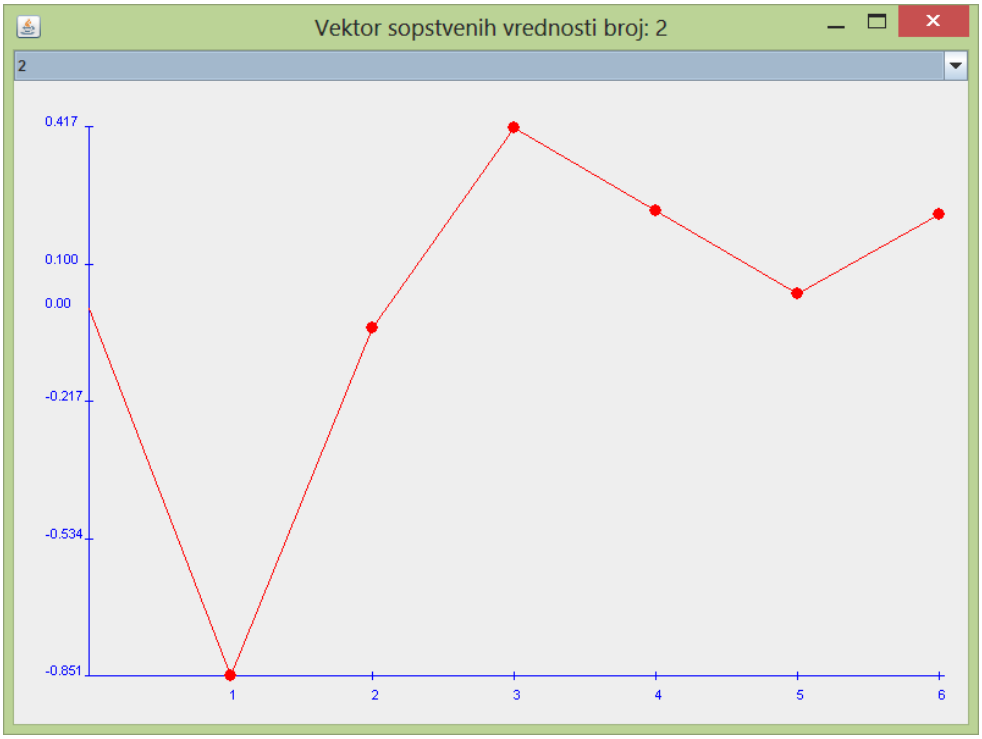
Slika 7.5 Sopstvene vrednosti Laplasove matrice

Spektralna klasterizacija se zasniva na analizi sopstvenog vektora koji odgovara drugoj najmanjoj sopstvenoj vrednosti. Vrednostisopstvenog vektora koji odgovara drugoj najmanjoj sopstvenoj vrednostisu prikazane na slici 7.6:



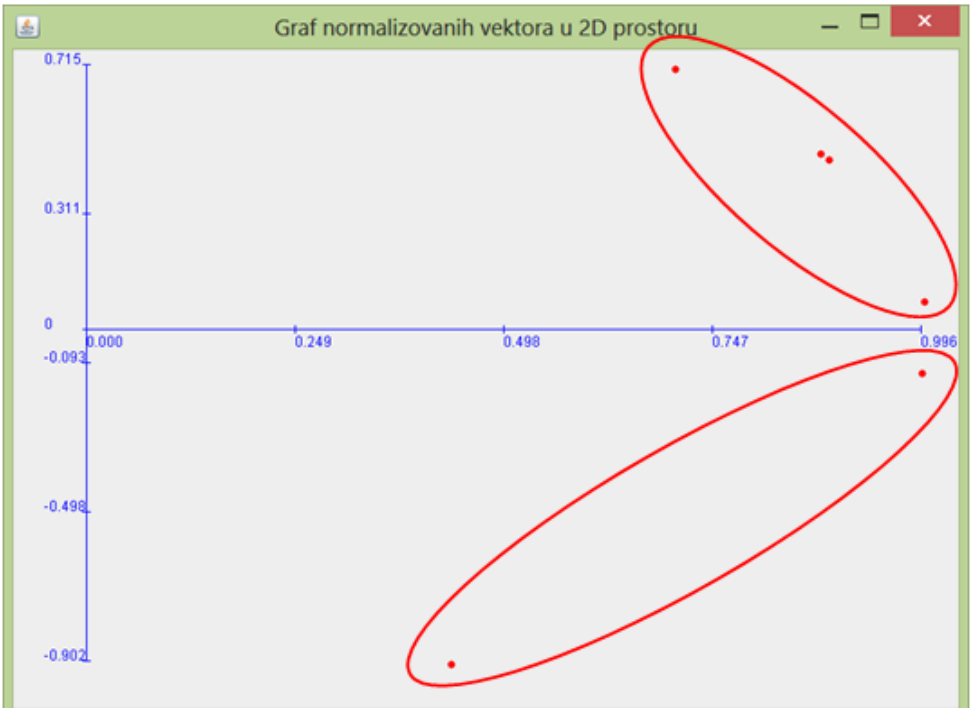
Slika 7.6 Sopstveni vektori

Njegov grafički prikaz je dat na slici 7.7:



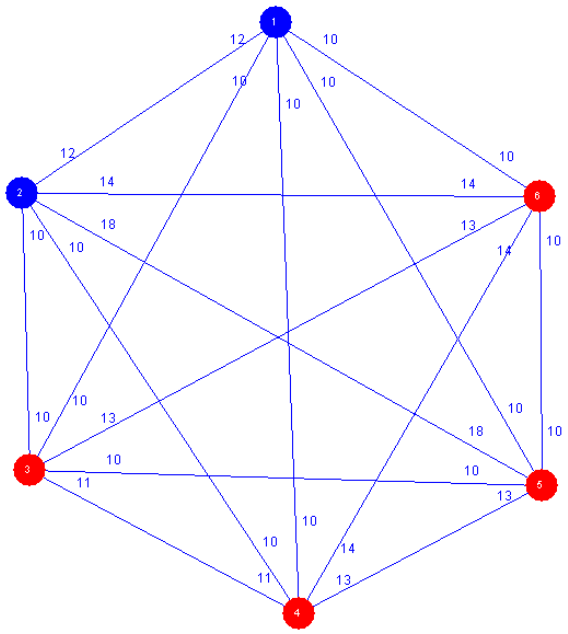
Slika 7.7 Grafički prikaz vektora sopstvenih vrednosti

Preslikavanje normalizovanih vrednosti prva dva sopstvena vektora Laplasove matrice u 2D prostoru je prikazano na slici 7.8:



Slika 7.8 Normalizovane vrednosti prva dva sopstvena vektora Laplasove matrice

Spektralna klasterizacija se zasniva na posmatranju sopstvenog vektora koji odgovara drugoj najmanjoj sopstvenoj vrednosti (Slika 7.6). Čvorovi koji odgovaraju pozitivnim vrednostima sopstvenog vektora pridružuju se prvom klasteru, a čvorovi koji odgovaraju negativnim vrednostima sopstvenog vektora pridružuju se drugom klasteru.



Slika 7.9 Klasterizacija čvorova

Rezultat klasteringa može se videti direktno sa grafa, na kome su čvorovi koji pripadaju prvom klasteru prikazani crvenom bojom, a čvorovi koji pripadaju drugom klasteru plavom bojom (Slika 7.9).

8. EVALUACIJA PREDLOŽENOG REŠENJA

Unapređenje učenja primenom algoritama WMN protokola rutiranja ogleda se u prednostima koje pružaju servisi zasnovani na računarstvu u oblaku i virtualizacija. Ti servisi omogućavaju:

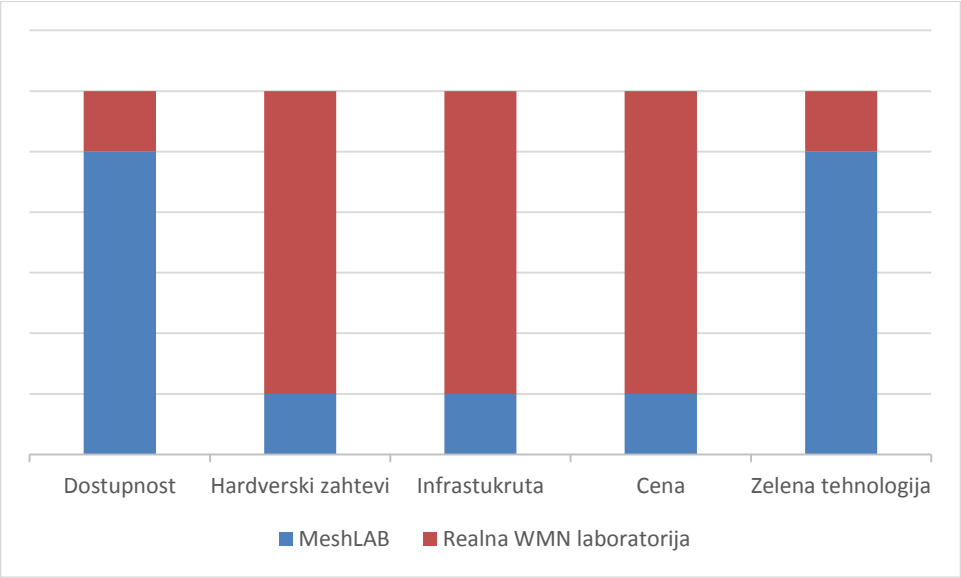
- dostupnost okruženja MeshLAB
- male hardverske zahtevi na strani klijenata (studenata)
- nema ulaganja u infrastrukturu i
- okruženje spada u zelene tehnologije.

U tabeli 8.1 prikazane su prednosti u performansama učenja bežičnih mesh mreža pomoću MeshLAB okruženja.

	MeshLAB	Realna WMN laboratorija
Dostupnost	+	-
Hardverski zahtevi	+	-
Infrastukruta	+	-
Cena	+	-
Zelena tehnologija	+	-

Tabela 8.1 Prednosti MeshLAB okruženja

Na grafikonu 8.1 vidi se da su prednosti ovakvog pristupa za učenje i testiranje algoritama za mobilne i mesh mreže izražene. Prednosti se ogledaju u dostupnosti okruženja koje je instalirano na VPS serveru (Virtual Private Server) koji se hostuje na nekom od provajdera VPS servisa (Amazon, Google, Oracle itd.). Hardverski zahtevi na strani klijenata su web pregledač za osnovnu upotrebu i ssh klijent za napredne korisnike. Infrastruktura za jednu laboratoriju ovakvog tipa koja podržava instalaciju 100 WMN čvorova košta 5$ mesečno. S druge strane, realna WMN laboratorija dostupna je samo na delu gde je fizički instalirana, hardverski zahtevi su mnogo veći u odnosu na virtualnu laboratoriju, a cena jedne realne WMN mreže može da bude i nekoliko desetina hiljada dolara.



Grafikon 8.1

Tabela 8.2 prikazuje nedostatke MeshLAB okruženja, koji se ogledaju u nedostatku testiranja različitih protokola rutiranja za koje postoji podrška, ali proces implementacije zahteva vreme. Takođe, postoji mnogo tipova mobilnih mreža, pa je potrebno raditi na prilagođavanju MeshLAB okruženja u odnosu na protokol ili tip mreže.

	OLSR	B.A.T.M.A.N.	Babel	BMx6
MeshLAB	dostupan	potrebna implementacija	potrebna implementacija	potrebna implementacija
Realna WMN laboratorija	dostupan	dostupan	dostupan	dostupan

Tabela 8.2 Nedostaci MeshLAB okruženja

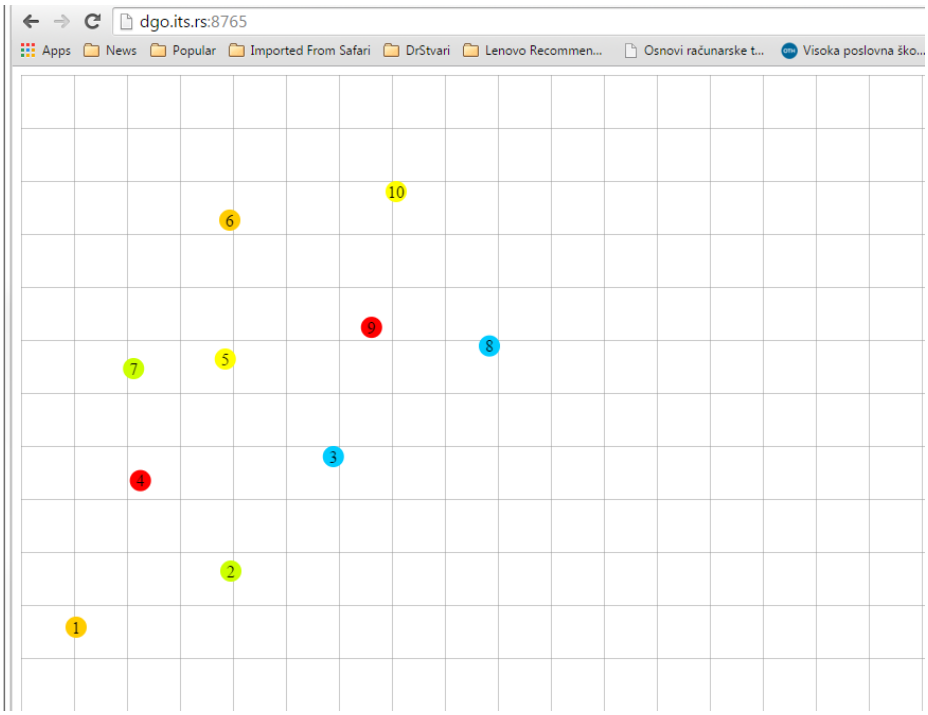
U tabeli 8.3 prikazani su ukupni utisci studenata o MeshLAB emulatoru, s prosečnim vrednostima iz svih dobijenih anketa.

	Ocena
Upotreba MeshLAB okruženja (1=teško, 5=lako)	2,8
Razumevanje WMN mreža i usmeravanja paketa primenom OLSR protokola u odnosu na materijale za učenje koji nisu uključeni u okruženje.	4,6

Tabela 8.3 Utisci studenata

Kako bi se izvršila evaluacija korišćenja MeshLAB okruženja na proces učenja apstraktnih pojmova algoritama za rutiranje iz predmeta Bežične mreže u odnosu na realnu laboratoriju za mesh mreže, testirana je propusna moć između datih čvorova kreirane laboratorije uporedo s realnim uređajima s istim protokolom za usmeravanje.

Kreirana je mreža WMN čvorova (Slika 8.1)



Slika 8.1 Mreža WMN čvorova

Korišćenjem alata iperf -sui1 testiran je protok prema čvoru 2, a kasnije i prema najudaljenijem čvoru 10.

Prikaz rezultata čvor 2 prema čvoru 1:

```
root@dgo:~# docker exec -i node1 telnet localhost 2004

root@7fa0ed9fab2a:/# iperf -uc 10.0.0.1 -b 50M -t 10

------------------------------------------------------------

Client connecting to 10.0.0.1, UDP port 5001

Sending 1470 byte datagrams

UDP buffer size:  208 KByte (default)

------------------------------------------------------------

[  3] local 10.0.0.2 port 38813 connected with 10.0.0.1 port 5001
```

```
[ ID] Interval       Transfer     Bandwidth

[  3]  0.0-10.0 sec  59.5 MBytes  49.9 Mbits/sec

[  3] Sent 42423 datagrams

[  3] Server Report:

[  3]  0.0-10.0 sec  8.44 MBytes  7.07 Mbits/sec   0.933 ms 36400/424222
(86%)

[  3]  0.0-10.0 sec  1 datagrams received out-of-order
```

Prikaz rezultata na čvoru 1:

```
[  4] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 38813

[  4]  0.0- 1.0 sec  1.58 MBytes  13.3 Mbits/sec   0.849 ms 2265/ 3393 (67%)

[  4]  1.0- 2.0 sec   765 KBytes  6.27 Mbits/sec   0.476 ms 4468/ 5001 (89%)

[  4]  2.0- 3.0 sec   751 KBytes  6.15 Mbits/sec   0.364 ms 3734/ 4257 (88%)

[  4]  3.0- 4.0 sec   777 KBytes  6.36 Mbits/sec   0.600 ms 3707/ 4248 (87%)

[  4]  4.0- 5.0 sec   775 KBytes  6.35 Mbits/sec   0.474 ms 3701/ 4241 (87%)

[  4]  5.0- 6.0 sec   778 KBytes  6.37 Mbits/sec   0.474 ms 3704/ 4246 (87%)

[  4]  6.0- 7.0 sec   798 KBytes  6.54 Mbits/sec   0.334 ms 3697/ 4253 (87%)

[  4]  7.0- 8.0 sec   795 KBytes  6.52 Mbits/sec   0.340 ms 3673/ 4227 (87%)

[  4]  8.0- 9.0 sec   787 KBytes  6.44 Mbits/sec   0.364 ms 3729/ 4277 (87%)

[  4]  9.0-10.0 sec   791 KBytes  6.48 Mbits/sec   0.378 ms 3669/ 4220 (87%)

[  4]  0.0-10.0 sec  1 datagrams received out-of-order

[  4]  0.0-10.0 sec  8.44 MBytes  7.07 Mbits/sec   0.934 ms 36400/42422 (86%)
```

Testiranje najudaljenijeg čvora 10 prema čvoru 1:

```
root@4f25ba79518c:/# iperf -uc 10.0.0.1 -b 50M -t 10

------------------------------------------------------------

Client connecting to 10.0.0.1, UDP port 5001

Sending 1470 byte datagrams

UDP buffer size:  208 KByte (default)
```

```
------------------------------------------------------------
[  3] local 10.0.0.9 port 53760 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  58.9 MBytes  49.4 Mbits/sec
[  3] Sent 42050 datagrams
[  3] Server Report:
[  3]  0.0-10.2 sec  3.78 MBytes  3.09 Mbits/sec  15.724 ms 39352/42045 (94%)
[  3]  0.0-10.2 sec  1 datagrams received out-of-order
```

Prikaz rezultata na čvoru 1

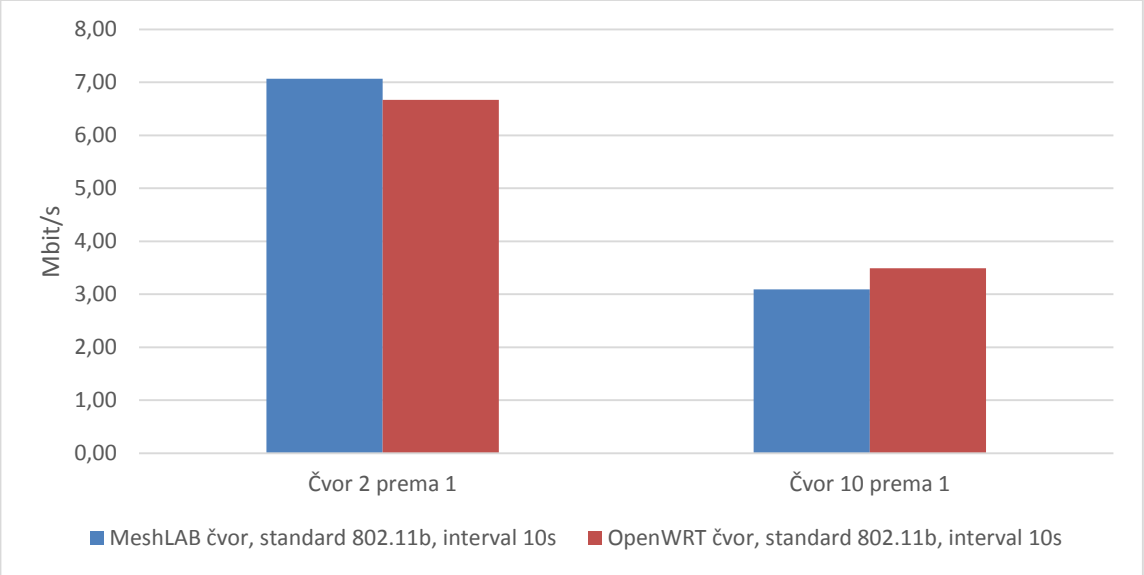
```
[ 4] local 10.0.0.1 port 5001 connected with 10.0.0.9 port 53760
[ 4]  0.0- 1.0 sec  685 KBytes  5.61 Mbits/sec  1.129 ms 3596/ 4073 (88%)
[ 4]  1.0- 2.0 sec  346 KBytes  2.83 Mbits/sec  1.365 ms 3951/ 4192 (94%)
[ 4]  2.0- 3.0 sec  312 KBytes  2.55 Mbits/sec   1.358 ms 3621/ 3838 (94%)
[ 4]  3.0- 4.0 sec  334 KBytes  2.74 Mbits/sec   0.324 ms 4264/ 4497 (95%)
[ 4]  4.0- 5.0 sec  444 KBytes  3.63 Mbits/sec   1.936 ms 3914/ 4223 (93%)
[ 4]  5.0- 6.0 sec  356 KBytes  2.92 Mbits/sec   1.201 ms 4038/ 4286 (94%)
[ 4]  6.0- 7.0 sec  355 KBytes  2.90 Mbits/sec   0.959 ms 3991/ 4238 (94%)
[ 4]  7.0- 8.0 sec  201 KBytes  1.65 Mbits/sec   1.853 ms 2207/ 2347 (94%)
[ 4]  8.0- 9.0 sec  527 KBytes  4.32 Mbits/sec   1.353 ms 5710/ 6077 (94%)
[ 4]  9.0-10.0 sec  301 KBytes  2.47 Mbits/sec   0.738 ms 4031/ 4241 (95%)
[ 4]  0.0-10.2 sec 3.78 MBytes  3.09 Mbits/sec  15.725 ms 39352/42045 (94%)
[ 4]  0.0-10.2 sec  1 datagrams received out-of-order
```

U tabeli 8.4 dat je prikaz propusne moći u odnosu na realne WMN čvorove.

	Čvor 2→1	Čvor 10→1
MeshLAB čvor, standard 802.11b, interval 10s	7.07 Mbits/sec	3.09 Mbits/sec
OpenWRT čvor, standard 802.11b, interval 10s	6.67 Mbits/sec	3.49 Mbits/sec

Tabela 8.4 Uporedni prikaz propusne moći

Takođe, grafički se može prikazati propusna moć u odnosu na realne WMN čvorove (Grafikon 8.2).



Grafikon 8.2

Primenom matematičkog modela iz spektralne teorije grafova (A. Zakic et al., 2014) istraživačima je omogućena analiza različitih protokola u domenu mobilnih mesh mreža. Primer klasterizacije čvorova dat u Poglavlju 7. jeste rezultat analize mesh protokola u slučaju kad postoji više hiljada čvorova u mreži. Ovaj model otvara mogućnost za buduća istraživanja u domenu optimizacije usmeravanja paketa u mesh mrežama.

Prikupljeni podaci pokazuju značaj upotrebe emulatora kao optimalnog rešenja za unapređenje performansi obrazovnih sistema u svim segmentima edukacije studenata. Ovaj pristup omogućava studentima da pored kurseva, materijala i vežbi, savladaju veštine koristeći emulatore kao zamenu za realne sisteme.

Primarna prednost upotrebe MeshLAB okruženja nije samo u okviru kursa Bežične mreže. To je koncept koji može da posluži u kreiranju sličnih distribuiranih aplikacija u domenu web programiranja, administracije internet servisa (email, www, DNS itd.), upravljanje podacima, elektronskog poslovanja itd.

Suština unapređenja performansi obrazovnih sistema ogleda se u izgradnji distribuiranih aplikacija primenom savremenih tehnologija. Za izgradnju ovakvog okruženja potrebno je da se studenti i predavači aktivno uključe u proces razvoja novih tehnologija, da bi stvorili optimalne uslove za eksperimente i nova naučna ostvarenja u oblasti informacionih i komunikacionih tehnologija.

9. ZAKLJUČAK

WMN predstavlja potpuno nov koncept i zbog prirode bežičnih i MANET (Mobile Ad hoc Networks) mreža, testiranje performasi protokola za rutiranje postao je glavni problem. Suština funkcionalnosti mesh mreža jeste u rutiranju paketa (Routing) pomoću koga se omogućava primanje i slanje podataka bilo gde u mreži. Protokoli rutiranja (Routing Protocol) u mesh mrežama omogućavaju čvorištima funkcionalnu i optimalnu putanju kroz mrežu. U odnosu na tradicionalne bežične LAN mreže (WLAN) i mreže za mobilnu telefoniju, bežična mesh mreža (WMN, Wireless Mesh Network) predstavlja pomak jer se lako nadograđuje i širi. Zbog brojnih prednosti u odnosu na tradicionalne mreže, WMN su budućnost bežičnih mreža. Otpornost na kvarove i jednostavna nadogradnja su glavne prednosti WMN mreža. Sama topologija, odnosno veliki broj čvorova u mreži čini WMN mreže otpornim na kvarove. Prednost ove tehnologije jeste ta što se veoma lako nadograđuje postojeća mrežna infrastruktura. Takođe, osobine ove tehnologije umrežavanja su jednostavnost proširenja, razvoja i nadogradnje, kao i pouzdanost s malim brojem prekida. Većina mesh mreža je kompatabilna s ostalim mrežnim tehnologijama, što čini sigurnosne mehanizme jednakim. Konvencionalni WLAN sigurnosni mehanizmi (WPA2/802.11i) pružaju standardne metode autentifikacije, kontrole pristupa i enkripcije između korisnika i pristupne tačke (Access Point, AP). Uprkos tome, postoji više vrsta mesh mrežnih arhitektura, koje podrazumevaju različit pristup mreži, a time i različit pristup sigurnosti mesh mreže. U budućnosti će sve mesh mreže imati standardizovane sigurnosne mehanizme zasnovane na 802.11s standardu sigurnosti računarskih mreža. Mehanizmi se odnose na korisnika, ad hok mrežu, kao i centralnu mesh mrežu na koju je ad hok mreža povezana. Postoji veliki broj protokola rutiranja u ad hok mesh mrežama, a njihov dizajn se još uvek istražuje kako bi se ubrzao prenos podataka. Istraživanje o ključnim teorijama za rutiranje paketa u WMN i teorijski i empirijski je od velikog značaja.

U ovoj tezi se analizira WMN protokola rutiranja, modelovanje i implementacija softverskog okruženja za emulaciju čvorova WMN mreže u edukativnom domenu (MeshLAB). U cilju približavanja uslova testiranja različitih WMN protokola rutiranja s realnim uređajima u razvoju softverskog okruženja korišćeni su računarstvo u oblaku (Cloud computing), virtualizacija na OS nivou, hardverska virtualizacija, virtualni privatni serveri (VPS – Virtual Private Server). Implementacijja okruženja zasniva se na Linux operativnom sistemu, a kod je pisan u programskim jezicima Python, Go lang, C++ i Bourne shell.

Primena virtualizacije gde se gost OS izvršava potpuno nezavisno omogućava emulaciju WMN čvorova i u zavisnosti od potrebe moguće je izabrati operativni sistem ili platformu koja je raspoloživa i koja podržava WMN protokole rutiranja (OLSR., Babel,

B.A.T.M.A.N.). Softverski sistemi zasnovani na računarstvu u oblaku omogućavaju projektovanja emulatora za testiranje protokola rutiranja (ova hipoteza je dokazana u Šestom poglavlju). Korišćenjem alata za upravljanje linux kontejnerima kreirani su čvorovi sa unapred definisanim protokolima, u ovom slučaju OLSR. Na ovaj način, u zavisnosti od količine RAM memorije i procesorske moći servera, moguće je kreirati i više hiljada čvorova. Ovaj model omogućava: smanjenje troškova pri testiranju novih uređaja, aplikacija ili protokola u ovom slučaju, analizu dobijenih rezultata, prilagođavanje okruženja zahtevima koji su potrebni u kreiranju nekog modela, sposobnost simulacije različitih arhitektura i tipova hardvera, sposobnost simulacije različitih platformi (Android, iOS, OpenWRT, Windows Mobile);

Karakteristike realizovanog sistema prikazane su matematičkim modelom iz spektralne teorije grafova, čime je u Sedmom poglavlju dokazana hipoteza da se definisanjem matematičkog modela studentima omogućava lakši pristup apstraktnim pojmovima iz spektralne teorije grafova na čijim teoremama se zasnivaju mesh mreže. Opisuje se primena matrice susedstva i Laplasove teoreme za kreiranje matematičkog modela na osnovu parametara dobijenih iz vežbi kreiranih pomoću emulatora MeshLAB (edukativnog modela za kreiranje WMN mreže). Proučavanjem primene spektralne teorije grafova u WMN mrežama stvorila se ideja o klasterizaciji čvorova, koja je obrazložena u disertaciji, i koja bi u budućim istraživanjima mogla da ima veliki doprinos u razvoju protokola rutiranja u WMN mrežama.

Formirana je metodologija projektovanja softverskog okruženja za emulaciju čvorova WMN mreže MeshLAB, opisan je generalni model WMN mreža, protokoli za rutiranje, dat je osvrt na bezbednosne protokole u WMN mrežama. Osnova okruženja jeste emulator koji omogućava testiranje različitih WMN protokola u edukativnom modelu.

U tezi su predstavljene osnovne osobine okruženja MeshLAB, kao i prva iskustva u upotrebi kroz primere. Cilj je da se omoguće dalja istraživanja WMN protokola rutiranja, da se studentima objasne apstraktni pojmovi mesh topologije kroz laboratorijske vežbe i primenom matematičkog modela iz spektralne teorije grafova na čijim teoremama se zasnivaju protokoli rutiranja u WMN mrežama.

Na osnovu prethodno navedenih činjenica dokazana je generalna hipoteza koja glasi: Novi način za povezivanje apstraktnih koncepata jeste korišćenje emulatora i virtualizacije za simulaciju rada realnih sistema razvijenih posebno za tu namenu.

Sa MeshLAB okruženjem studentima je kroz vizualizaciju olakšano razumevanje WMN tehnologije i MANET mreža, a predavačima i istraživačima olakšano kreiranje novih scenarija za testiranje protokola rutiranja u proizvoljnim uslovima. Samim tim je u Šestom poglavlju dokazana hipoteza da se pri razjašnjavanju apstraktnih pojmova na predavanjima nastavnici susreću s problemom testiranja različitih WMN protokola rutiranja, s velikim brojem čvorova.

Kritičkim pregledom postojećih istraživanja utvrđeno je da mrežni simulatori (ns-2, ns-3, OMNet i sl.) često daju različite rezultate prilikom testiranja protokola u odnosu na performanse realnih uređaja.

Doprinos izložene doktorske disertacije u domenu je analize i sinteze softverskog sistema koji treba da omogući kreiranje okruženja za WMN mreže pogodnog za rad u edukativnom domenu. Kao sastavni delovi doktorske disertacije sadržani su sledeći naučni doprinosi:

- sistematizacija i klasifikacija postojećih rešenja pri projektovanju emulatora iz oblasti WMN mreža;
- kreiranje metodologije projektovanja emulatora WMN mreža na osnovu klasifikovanih rešenja koja su bila primenjena, koja se primenjuju ili koja se mogu primeniti u okviru ove discipline;
- osnovni doprinos je predlog i implementacija novog edukativnog okruženja koji na jednostavan način treba da omogući kreiranje WMN mreža sposobnih za rad u edukativnom okruženju, za testiranje protokola rutiranja. Sistem obezbeđuje vizuelni pregled i izgradnju kompleksnog sistema WMN mreža od najnižeg nivoa, kao i mogućnostima povezivanja proizvoljnih modula korišćenjem alata;
- postavljeni analitički model opisuje ponašanje emulatora WMN mreža razvijenih prema predloženoj metodologiji prilikom rada u edukativnom okruženju, kao i matematički model primenom spektralne teorije grafova.

Prikazan je prototip emulatora razvijen prema opisanom postupku s ciljem da se omogući njegova primena u nastavi i u budućim istraživanjima WMN protokola rutiranja. Definisane su i laboratorijske vežbe koje su implementirane u softverskom okruženju. Dat je analitički model koji opisuje karakteristike realizovanog sistema prikazan matematičkim modelom iz spektralne teorije grafova.

Budući radovi se zasnivaju na upotrebi okruženja MeshLAB za testiranje performansi različitih WMN protokola (OLSR., Babel, B.A.T.M.A.N.), kao i na primeni klasterizacije iz spektralne teorije grafova za optimizaciju usmeravanja paketa kroz WMN ili MANET mrežu koja sadrži veliki broj mobilnih čvorova.

LITERATURA

A. Neumann and C. Aichele and M. Lindner and S. Wunderlich. (2008). Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.). Internet draft, work in progress.

Abolhasan, M., Hagelstein, B., & Wang, J. P. (2009, October). Real-world performance of current proactive multi-hop mesh protocols. In*Communications, 2009. APCC 2009. 15th Asia-Pacific Conference on* (pp. 44–47). IEEE.

Adjih, C., Baccelli, E., Clausen, T. H., Jacquet, P., & Rodolakis, G. (2004). Fish eye OLSR scaling properties. *Communications and Networks, Journal of*, *6*(4), 343-351.

Ahrenholz, J., Danilov, C., Henderson, T. R., & Kim, J. H. (2008, November). CORE: A real-time network emulator. In *Military Communications Conference, 2008. MILCOM 2008. IEEE* (pp. 1–7). IEEE.

Akyildiz, I. F., & Wang, X. (2005). A survey on wireless mesh networks. *Communications Magazine, IEEE*, *43*(9), S23–S30.

Akyildiz, I. F., Wang, X., & Wang, W. (2005). Wireless mesh networks: a survey. *Computer networks*, *47*(4), 445–487.

Akyildiz, I., & Wang, X. (2009). *Wireless mesh networks* (Vol. 3). John Wiley & Sons.

Alotaibi, E., & Mukherjee, B. (2012). A survey on routing algorithms for wireless Ad-Hoc and mesh networks. *Computer Networks*, *56*(2), 940–965.

Al-Shurman, M., Yoo, S. M., & Park, S. (2004, April). Black hole attack in mobile ad hoc networks. In *Proceedings of the 42nd annual Southeast regional conference* (pp. 96–97). ACM.

Andreas Tonnesen and Thomas Lopatic and Hannes Gredler and Bernd Petrovitsch and Aaron Kaplan and Sven-Ola Tucke, (2009). OLSR development at http://www.olsr.org/, Internet: http://www.olsr.org/.

Ashraf, U., Juanole, G., & Abdellatif, S. (2007, October). Evaluating routing protocols for the wireless mesh backbone. In *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on* (pp. 40). IEEE.

Awerbuch, B., Curtmola, R., Holmer, D., Rubens, H., & Nita-Rotaru, C. (2005, September). On the survivability of routing protocols in ad hoc wireless networks. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on* (pp. 327–338). IEEE.

Awerbuch, B., Holmer, D., Nita-Rotaru, C., & Rubens, H. (2002, September). An on-demand secure routing protocol resilient to byzantine failures. In*Proceedings of the 1st ACM workshop on Wireless security* (pp. 21–30). ACM.

Bahr, M. (2006, August). Proposed routing for IEEE 802.11 s. WLAN mesh networks. In *Proceedings of the 2nd annual international workshop on Wireless internet* (p. 5). ACM.

Bahr, M. (2007, October). Update on the Hybrid Wireless Mesh Protocol of IEEE 802.11 s. In *MASS* (pp. 1–6).

Bakht, H. (2011). Survey of routing protocols for mobile ad-hoc network. *International Journal of Information and Communication Technology Research*, *1*(6).

Barolli, L., Ikeda, M., De Marco, G., Durresi, A., & Xhafa, F. (2009, May). Performance analysis of OLSR and B.A.T.M.A.N. protocols considering link quality parameter. In *Advanced Information Networking and Applications, 2009. AINA'09. International Conference on* (pp. 307–314). IEEE.

Basagni, S., Chlamtac, I., Syrotiuk, V. R., & Woodward, B. A. (1998, October). A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking* (pp. 76–84). ACM.

Blom, R. (1985, January). An optimal class of symmetric key generation systems. In *Advances in cryptology* (pp. 335-338). Springer Berlin Heidelberg.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y. C., & Jetcheva, J. (1998, October). A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking* (pp. 85–97). ACM.

Brown, T. X., James, J. E., & Sethi, A. (2006, May). Jamming and sensing of encrypted wireless ad hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing* (pp. 120–130). ACM.

Bruno, R., Conti, M., & Gregori, E. (2005). Mesh networks: commodity multihop ad hoc networks. *Communications Magazine, IEEE*, *43*(3), 123–131.

Chroboczek, J. (2011). The babel routing protocol.

Clausen, T. H., Adjih, C., Dearlove, C. M., & Dean, J. W. (2009). Generalized MANET packet/message format.

Clausen, T., & Jaqcquet, P. (2003). Optimized link state routing (OLSR) RFC 3626. *IETF Networking Group (October 2003)*.

Clausen, T., Dearlove, C., Jacquet, P., & Herberg, U. (2006). The optimized link state routing protocol version 2. *draft-ietf-manet-olsrv2-00, Work in progress*.

Curtmola, R., & Nita-Rotaru, C. (2009). BSMR: byzantine-resilient secure multicast routing in multihop wireless networks. *Mobile Computing, IEEE Transactions on*, *8*(4), 445–459.

Cvetkovic, D., & Gutman, I. Applications of graph spectra. *Zbornik radova (Beograd)*, (2009), *13*, 21.

Cvetkovic, D., Doob, M., & Sachs, H. (1995). Spectra of Graphs-Theory and Applications, III revised and enlarged edition. *Johan Ambrosius Bart Verlag, Heidelberg-Leipzig.*

Das, S. R., Belding-Royer, E. M., & Perkins, C. E. (2003). Ad hoc on-demand distance vector (AODV) routing.

De Couto, D. S., Aguayo, D., Bicket, J., & Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, *11*(4), 419–434.

Dobrynin, A. A., Entringer, R., & Gutman, I. (2001). Wiener index of trees: theory and applications. *Acta Applicandae Mathematica*, *66*(3), 211–249.

Dong, J., Curtmola, R., & Nita-Rotaru, C. (2009). Secure network coding for wireless mesh networks: Threats, challenges, and directions. *Computer Communications*, *32*(17), 1790–1801.

Eriksson, J., Krishnamurthy, S. V., & Faloutsos, M. (2006, November). Truelink: A practical countermeasure to the wormhole attack in wireless networks. In *Network Protocols, 2006. ICNP'06. Proceedings of the 2006 14th IEEE International Conference on* (pp. 75–84). IEEE.

Evseev, S. P., Dorohov, A. V., & Korolj, O. G. (2011). Mehanizmi zaštite informacija u kompjuterskim mrežama i sistemima. *Vojnotehnički glasnik*, *59*, 15–39.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, *23*(2), 298–305.

Gallier, J. (2014). Spectral Theory of Unsigned and Signed Graphs Applications to Graph Clustering: a Survey.

Gast, M. (2005). *802.11 wireless networks: the definitive guide*. "O'Reilly Media, Inc."

Gupta, J., Bedi, P. K., & Gupta, N. (2011). Fault Tolerant Wireless Mesh Network: An Approach. *International Journal of Computer Applications*, *23*(3).

Hiertz, G. R., Denteneer, D., Max, S., Taori, R., Cardona, J., Berlemann, L., & Walke, B. (2010). IEEE 802.11 s: the WLAN mesh standard. *Wireless Communications, IEEE*, *17*(1), 104–111.

Hu, Y. C., Johnson, D. B., & Perrig, A. (2003). SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad hoc networks*, *1*(1), 175–192.

Hu, Y. C., Perrig, A., & Johnson, D. B. (2003, April). Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies* (Vol. 3, pp. 1976–1986). IEEE.

Hu, Y. C., Perrig, A., & Johnson, D. B. (2003, September). Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2nd ACM workshop on Wireless security* (pp. 30–40). ACM.

Hu, Y. C., Perrig, A., & Johnson, D. B. (2005). Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless networks*, *11*(1-2), 21-38.

IEEE 802.11 Working Group. (2010). IEEE Standard for Information Technology–Telecommunications and information exchange between systems – Local and metropolitan area networks–Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Wireless Access in Vehicular Environments. *IEEE Std*, *802*, 11p.

IEEE Standards Association. Working Group P1876-Networked Smart Learning Objects for Online Laboratories, 2012. *URL: http://ieeesa. centraldesktop. Com*.

Information Processing Systems – Open Systems Interconnection – Basic Reference Model, ISO/Draft International Standard (DIS) 7498 (ISO/TC97/SC16 N890), February 1982, as revised by: Changes to DIS 7498, (ISO/TC97/SC16 N1226), June 1982.

Jacquet, P., Laouiti, A., Minet, P., & Viennot, L. (2002). Performance of multipoint relaying in ad hoc mobile routing protocols. In *NETWORKING 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications* (pp. 387–398). Springer Berlin Heidelberg.

Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International* (pp. 62–68). IEEE.

Jayakumar, G., & Gopinath, G. (2007). Ad hoc mobile wireless networks routing protocols-a review. *Journal of Computer science*, *3*(8), 574–582.

Jing, X., & Lee, M. J. (2004). Energy-aware algorithms for AODV in ad hoc networks. *Proceedings of Mobile Computing and Ubiquitous Networking*, 466–468.

Johnson, D., Hu, Y., & Maltz, D. (2007). *The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4*. RFC 4728, February.

Johnson, D., Ntlatlapa, N., & Aichele, C. (2008). Simple pragmatic approach to mesh routing using B.A.T.M.A.N.

Kawadia, V., & Kumar, P. R. (2005). A cautionary perspective on cross-layer design. *Wireless Communications, IEEE*, *12*(1), 3–-11.

Khan, S., Mast, N., Loo, K. K., & Silahuddin, A. (2008). Passive security threats and consequences in IEEE 802.11 wireless mesh networks. *2; 3*.

Kiess, W., & Mauve, M. (2007). A survey on real-world implementations of mobile ad-hoc networks. *Ad Hoc Networks*, *5*(3), 324–339.

Kulla, E., Ikeda, M., Barolli, L., & Miho, R. (2010, November). Impact of source and destination movement on MANET performance considering B.A.T.M.A.N. and AODV protocols. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on* (pp. 94–101). IEEE.

Kulla, E., Ikeda, M., Barolli, L., Miho, R., & Kolici, V. (2010, September). Effects of Source and Destination Movement on MANET Performance Considering OLSR and AODV Protocols. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on* (pp. 510–515). IEEE.

Kuppusamy, P., Thirunavukkarasu, K., & Kalaavathi, B. (2011, April). A study and comparison of OLSR, AODV and TORA routing protocols in ad hoc networks. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on* (Vol. 5, pp. 143–147). IEEE.

Lashkari, A. H., Danesh, M. M. S., & Samadi, B. (2009, August). A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on* (pp. 48–52). IEEE.

Law, Y. W., Van Hoesel, L., Doumen, J., Hartel, P., & Havinga, P. (2005, November). Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks* (pp. 76–88). ACM.

Li, C., Wang, Z., & Yang, C. (2011). Secure Routing for Wireless Mesh Networks. *IJ Network Security*, *13*(2), 109–120.

Lorincz, J., & Begusic, D. (2006, February). Physical layer analysis of emerging IEEE 802.11 n WLAN standard. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference* (Vol. 1, pp. 6). IEEE.

Macker, J. P., & Dean, J. W. (2003, October). A study of link state flooding optimizations for scalable wireless networks. In *Military Communications Conference, 2003. MILCOM'03. 2003 IEEE* (Vol. 2, pp. 1262–1267). IEEE.

Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000, August). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 255–265). ACM.

Mathis, M., Mahdavi, J., Floyd, S., Romanow, A., & Options, T. S. A. (1996). RFC 2018. *Internet Engineering Task Force (IETF)*.

Mell, P., & Grance, T. (2009). The NIST definition of cloud computing. *National Institute of Standards and Technology*, *53*(6), 50.

Milovanovic, I. Z., Milovanovic, E. I., & Zakic, A. (2014). A short note on graph energy. *MATCH Commun. Math. Comput. Chem*, *72*, 179–182.

Mohar, B., & Alavi, Y. (1991). The Laplacian spectrum of graphs. Graph theory, *combinatorics, and applications*, *2*, 871–898.

Murray, D., Dixon, M., & Koziniec, T. (2010, October). An experimental comparison of routing protocols in multi hop ad hoc networks. In *Telecommunication Networks and Applications Conference (ATNAC), 2010 Australasian* (pp. 159–164). IEEE.

Narten, T., Simpson, W. A., Nordmark, E., & Soliman, H. (2007). Neighbor discovery for IP version 6 (IPv6).

Neumann, A., Aichele, C., Lindner, M., & Wunderlich, S. (2008). Better approach to mobile ad-hoc networking (B.A.T.M.A.N.). *IETF draft, October*.

Newsome, J., Shi, E., Song, D., & Perrig, A. (2004, April). The Sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (pp. 259–268). ACM.

Nguyen, D., & Minet, P. (2007, April). Scalability of the olsr protocol with the fish eye extension. In *Networking, 2007. ICN'07. Sixth International Conference on* (pp. 88-88). IEEE.

Ozugur, T., Copeland, J. A., Naghshineh, M., & Kermani, P. (1999). Next-generation indoor infrared LANs: issues and approaches. *Personal Communications, IEEE, 6*(6), 6–19.

Papadimitratos, P., & Haas, Z. J. (2003, January). Secure link state routing for mobile ad hoc networks. In *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on* (pp. 379–383). IEEE.

Paul, A. B., Konwar, S., Gogoi, U., Nandi, S., & Biswas, S. (2011, October). E-AODV for wireless mesh networks and its performance evaluation. In *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on* (pp. 26–33). IEEE.

Pei, G., Gerla, M., & Chen, T. W. (2000). Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on* (Vol. 1, pp. 70–74). IEEE.

Perkins, C. E. (2008). *Ad hoc networking*. Addison-Wesley Professional.

Perkins, C. E., & Bhagwat, P. (1994, October). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM Computer Communication Review* (Vol. 24, No. 4, pp. 234–244). ACM.

Perrig, A., Canetti, R., Tygar, J. D., & Song, D. (2005). The TESLA broadcast authentication protocol. *RSA CryptoBytes, 5*.

Pfaff, B., Pettit, J., Amidon, K., Casado, M., Koponen, T., & Shenker, S. (2009, October). Extending Networking into the Virtualization Layer. In *Hotnets*.

Ramaswamy, S., Fu, H., Sreekantaradhya, M., Dixon, J., & Nygard, K. E. (2003, June). Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks. In *international conference on wireless networks* (Vol. 2003).

Raniwala, A., & Chiueh, T. C. (2005, March). Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (Vol. 3, pp. 2223–2234). IEEE.

Roy, S., Addada, V. G. K., Setia, S., & Jajodia, S. (2005, September). Securing MAODV: attacks and countermeasures. In *SECON* (pp. 521–532).

Royer, E. M. (2000). Multicast ad hoc on-demand distance vector (MAODV) routing. *IETF Internet Draft, draft-ietf-manet-maodv-00. txt*.

Santivanez, C. A., & Ramanathan, R. (2004). Scalability of routing in ad hoc networks: principles and practice. In *Ad Hoc Wireless Networking* (pp. 561–621). Springer US.

Sanzgiri, K., Dahill, B., Levine, B. N., Shields, C., & Belding-Royer, E. M. (2002, November). A secure routing protocol for ad hoc networks. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on* (pp. 78–87). IEEE.

Scardamalia, M., & Bereiter, C. (2010). A brief history of knowledge building.Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie, 36(1).

Scheepers, M. J. (2014). Virtualization and Containerization of Application Infrastructure: A Comparison.

Sen, J., Chandra, M. G., Harihara, S. G., Reddy, H., & Balamuralidhar, P. (2007, December). A mechanism for detection of gray hole attack in mobile Ad Hoc networks. In *Information, Communications & Signal Processing, 2007 6th International Conference on* (pp. 1–5). IEEE.

Shah, S. I. H., & Shaheed, S. H. (2011). *Performance Evaluation of MANET Routing Protocols* (Doctoral dissertation, M. Sc. Thesis, Blekinge Institute of Technology, Sweden).

Staub, T., Gantenbein, R., & Braun, T. (2009, March). VirtualMesh: an emulation framework for wireless mesh networks in OMNeT++. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques* (p. 64). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Stevanović, D., Erić, M., & Starčević, D. (2004). A concept of ad hoc communication networks for the sensors networking in command control systems. *Vojnotehnički glasnik*, *52*(3-4), 379–389.

Tamilarasan, S. (2012). A Quantitative Study and Comparison of AODV, OLSR and TORA Routing Protocols in MANET. *International Journal of Computer Science Issues (IJCSI)*, *9*(1).

Tønnesen, A. (2004). Implementing and extending the optimized link state routing protocol. *Master's thesis. University of Oslo, Norway*.

Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization. James Turnbull.

Vassis, D., Kormentzas, G., Rouskas, A., & Maglogiannis, I. (2005). The IEEE 802.11 g standard for high data rate WLANs. *Network, IEEE*, *19*(3), 21–26.

Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio. An Updated Performance Comparison of Virtual Machines and Linux Containers. (2014). IBM Research Report.

Westcott, D. A., Coleman, D. D., Miller, B., & Mackenzie, P. (2011). *CWAP Certified Wireless Analysis Professional Official Study Guide: Exam PW0-270*. John Wiley & Sons.

Wood, A., & Stankovic, J. A. (2002). Denial of service in sensor networks.*Computer*, *35*(10), 54–62.

Yi, S., Naldurg, P., & Kravets, R. (2001, October). Security-aware ad hoc routing for wireless networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing* (pp. 299–302). ACM.

Zakic A., Veinovic M., Jovanovic N., T. S. (2015). LINUX CONTAINERS: DOCKER PLATFORM FOR DISTRIBUTED APPLICATIONS. *INFOTECH.*

Zakrzewska, A., Koszalka, L., & Pozniak-Koszalka, I. (2008, August). Performance study of routing protocols for wireless mesh networks. In *Systems Engineering, 2008. ICSENG'08. 19th International Conference on* (pp. 331–336). IEEE.

Zakrzewska, A., Koszalka, L., & Pozniak-Koszalka, I. (2008, August). Performance study of routing protocols for wireless mesh networks. In *Systems Engineering, 2008. ICSENG'08. 19th International Conference on* (pp. 331–336). IEEE.

Zapata, M. G., & Asokan, N. (2002, September). Securing ad hoc routing protocols. In Proceedings of the 1st ACM workshop on Wireless security (pp. 1–10). ACM.

Zhu, S., Xu, S., Setia, S., & Jajodia, S. (2006). LHAP: a lightweight network access control protocol for ad hoc networks. *Ad Hoc Networks*, *4*(5), 567–585.