

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA TELEKOMUNIKACIJE

NS-3 Epidemic routing

PROJEKTNİ ZADATAK IZ PREDMETA SIMULACIJA PROCESA U TELEKOMUNIKACIJSKIM MREŽAMA
PRAKTIČNI DIO

Ćutahija Zerina, 1685/17085
Hasanbegović Selma, 1574/17753
Mahovac Nerman, 1575/17919
Repeša Almin, 1684/17550
Velić Nejra, 1634/17313

Sarajevo, 2020. godina

Sadržaj

Sadržaj	i
1 Opis rješenja	1
2 Komparacija sa teoretskim rezultatima	6
2.1 Varijacija veličine raspona prenosa	6
2.2 Varijacija broja hop-ova	7
2.3 Varijacija veličine buffer-a	9
3 Diskusija i zaključak	11
Popis slika	12
Popis tablica	13
Literatura	14

1. Opis rješenja

Opis problema

Epidemic routing protokol jeste protokol za DTN, baziran na plavljenju (*flooding*). Svaki čvor u mreži propagira poruku/paket kroz mrežu tako što pravi replike istih i predaje ih svim čvorovima sa kojima je stupio u kontakt, analogno širenju zaraznih bolesti. Propagacija poruke se dešava sve dok poruka ne stigne na odredišni čvor ili dok se istekne rok za isporuku iste.

Analiza načina implementacije modela u NS-3

U ovom radu je korištena skripta *epidemic-benchmark.cc* preuzeta iz *Epidemic routing* foldera preuzetog sa <https://apps.nsnam.org/app/epidemic-routing/>. Svi rezultati prikazani u ovom radu poređeni su sa rezultatima datim u [1]. Simulirana je mreža od 50 čvorova, na području od 1500 m x 300 m. Odabrano je 45 čvorova da šalje poruke veličine 1KB prema preostala 44 čvora. Ukupan broj razmjenjenih poruka je $44 * 45 = 1980$ poruka(paketa), što je tabeli 1.1 označeno kao *Number of packets*. Veličina *buffer*-a je 2000, što u ovim scenarijima predstavlja beskonačnu veličinu. *Default*-na udaljenost za prenos podataka je postavljena na 50m. Model mobilnosti može biti statički *Grid* ili *RandomWayPoint*, u ovom slučaju izabran je *RandomWayPoint*. Kod ovog modela inicijalne pozicije su uniformne i slučajno raspodjeljene na području od 1500 m x 300 m. Podatkovni saobraćaj je generisan koristeći *OnOff* aplikacije i primljen na sinku (*PacketSink*).

U NS-3 simulatoru je najprije potrebno definisati parametre simulacije, pri čemu se definišu opći, aplikacijski te *Epidemic routing* parametri. Vrijednosti ovih parametara su prethodno objašnjene. Nakon toga potrebno je omogućiti prenos podatkovnog saobraćaja putem *OnOff* aplikacije i primanje tog saobraćaja na *PacketSink*-u, putem komandi:

LogComponentEnable ("OnOffApplication") i *LogComponentEnable ("PacketSink")*

Dalje je potrebno kreirati čvorove, tačnije kreira se "kontejner čvorova", koji ima predefinisani broj čvorova, u ovom slučaju je to 50. Kako je već prethodno rečeno model mobilnosti je *RandomWayPoint*, te se taj model mora instalirati na čvorove, uz dodavanje parametara koji su neophodni za ispravan rad datog modela. Pri tome se koristi komanda:

mobility.SetMobilityModel()

Slijedi podešavanje PHY i LL sloja, što se vrši postavljanjem različitih *helper*-a. Za potrebe epidemijskog rutiranja na spomenutim slojevima postavljaju se *WifiMAC helper*, *WifiPHY helper* te *WifiChannel helper*. Pri čemu se prva dva postavljaju na *default*-nu vrijednost. U *Wifi* kanalu se postavlja i propagaciono slabljene, korištenjem naredbe:

```
wifiChannel.AddPropagationLoss
```

Nakon toga se nizom komandi *epidemic.Set* postavljaju različite vrijednosti vezane za parametre *Epidemic routing* protokola, te se na osnovu tih parametara definiše epidemijsko rutiranje na samim čvorovima. Dalje je uređajima potrebno dodijeliti IP adresu i pod mrežnu masku, a nakon toga je potrebno izvršiti setup aplikacije i klijenta (*Application Setup* i *Client setup*, respektivno).

Tablica 1.1: Parametri simulacije [1]

Parametri	Vrijednost
Simulation area	300 m x 1500 m
Number of runs	100
Warmup time	10 s
Total simualtion time	200 000 s
Mobility model	Random waypoint
Node speed	0 - 20 m/s
Packet size	1024 B
Number of packets	1980 paket/simulacija
Link layer	wifib-11mbs
Propagation loss model	Range

Početak epidemijskog rutiranja se određuje naredbama:

```
Config::Connect("/NodeList/*/ApplicationList/*/ns3::OnOffApplication/Tx",  
MakeCallback(&SentPacket))
```

```
Config::Connect("/NodeList/*/ApplicationList/*/ns3::PacketSink/Rx",  
MakeCallback(&ReceivedPacket))
```

koje pozivaju funkcije *SentPacket* i *ReceivedPacket*. Pored toga poziva se i funkcija *Ratio()*. Navedene tri funkcije nisu inicijalno sadržane u korištenoj skripti *epidemic-routing.cc* već su uvedene zbog potrebe za očitavanjem vrijednosti kašnjenja i omjer (engl. *ratio*). Ovi parametri su koršteni za crtanje dijagrama, što će biti pokazano u narednom poglavlju, dok su funkcije *SentPacket*, *ReceivedPacket* i *Ratio()* preuzete iz skripte *sptm_test.cc* [2].

Sve skripte korištene za obavljanje navedenih funkcionalnosti kao i *bash* skripta, te skripta u kojoj je implementiran *gnuplot* za crtanje grafika, dati su na *github*-u, na linku: <https://github.com/sptmprojekatepidemic/SPTM-EPIDEMIC-ROUTING>, dok se preuzeti *Epidemic routing* folder može pronaći na [3].

Tablica 1.2: Parametri evaluacije *Epidemic Routing* protokola [1]

Parametri	Vrijednosti
HopCount	10
QueueLength	2 000
QueueEntityExpireTime	200 000
HostRecentPeriod	10
BeaconInterval	5 000
BeaconRandomness	5

Na slikama 1.1 i 1.2 dat je prikaz izgleda terminala prilikom pokretanja i simuliranja *Epidemic routing* protokola. Na prvoj slici prikazan je izgled terminala sa svim ispisanim postavkama, dok na drugoj slici vidimo da se paketi šalju i primaju na osnovu postavljениh parametara, kao što je dato u tabelama 1.1 i 1.2.

```

selma@selma-VirtualBox:~/ns-allinone-3.30.1/ns-3.30.1$ ./waf --run epidemic-benchmark
Waf: Entering directory `~/home/selma/ns-allinone-3.30.1/ns-3.30.1/build'
Waf: Leaving directory `~/home/selma/ns-allinone-3.30.1/ns-3.30.1/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.365s)
Number of wifi nodes: 50
Node speed: 10 m/s
Packet size: 1024 b
Transmission distance: 50 m
Hop count: 50
Queue length: 200 packets
Queue entry expire time: 1000 s
Beacon interval: 5 s
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.2 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.3 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.4 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.5 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.6 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.7 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.8 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.9 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.10 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.11 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.12 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.13 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.14 port 80 total Tx 1024 bytes
+11.000000000s 0 OnOffApplication:SendPacket(): At time 11s on-off application sent 1024 bytes to 10.1.1.15 port 80 total Tx 1024 bytes

```

Slika 1.1: Prikaz terminala

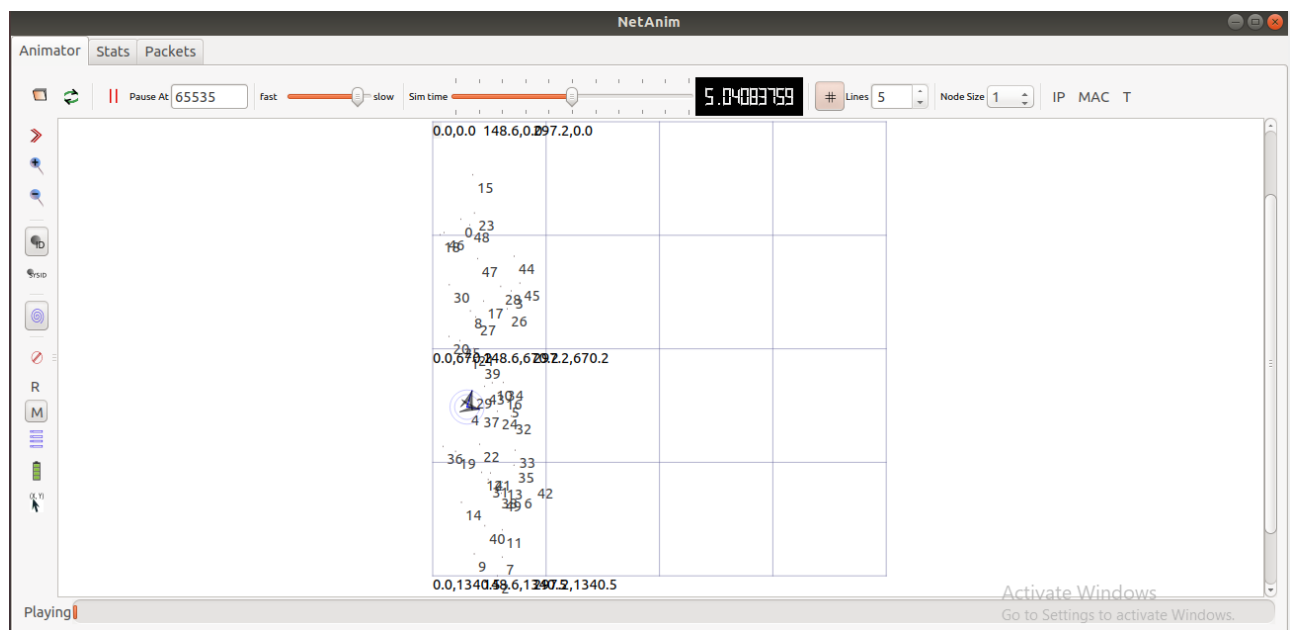
```

+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.35 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.36 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.37 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.38 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.39 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.40 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.41 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.42 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.43 port 80 total Tx 3072 bytes
+13.000000000s 44 OnOffApplication:SendPacket(): At time 13s on-off application sent 1024 bytes to 10.1.1.44 port 80 total Tx 3072 bytes
+15.091864162s 4 PacketsSink:HandleRead(): At time 15.0919s packet sink received 1024 bytes from 10.1.1.30 port 49157 total Rx 1024 bytes
+15.096465329s 1 PacketsSink:HandleRead(): At time 15.0965s packet sink received 1024 bytes from 10.1.1.26 port 49154 total Rx 1024 bytes
+15.120256939s 5 PacketsSink:HandleRead(): At time 15.1203s packet sink received 1024 bytes from 10.1.1.17 port 49158 total Rx 1024 bytes
+15.120276952s 12 PacketsSink:HandleRead(): At time 15.1203s packet sink received 1024 bytes from 10.1.1.42 port 49165 total Rx 1024 bytes
+15.160961404s 16 PacketsSink:HandleRead(): At time 15.161s packet sink received 1024 bytes from 10.1.1.6 port 49168 total Rx 1024 bytes
+15.173163705s 24 PacketsSink:HandleRead(): At time 15.1732s packet sink received 1024 bytes from 10.1.1.33 port 49177 total Rx 1024 bytes
+15.173667017s 8 PacketsSink:HandleRead(): At time 15.1737s packet sink received 1024 bytes from 10.1.1.18 port 49161 total Rx 1024 bytes
+15.177710166s 29 PacketsSink:HandleRead(): At time 15.1777s packet sink received 1024 bytes from 10.1.1.5 port 49181 total Rx 1024 bytes
+15.178161842s 25 PacketsSink:HandleRead(): At time 15.1782s packet sink received 1024 bytes from 10.1.1.2 port 49177 total Rx 1024 bytes
+15.197137712s 40 PacketsSink:HandleRead(): At time 15.1971s packet sink received 1024 bytes from 10.1.1.10 port 49192 total Rx 1024 bytes
+15.203637762s 17 PacketsSink:HandleRead(): At time 15.2036s packet sink received 1024 bytes from 10.1.1.9 port 49169 total Rx 1024 bytes
+15.232675388s 4 PacketsSink:HandleRead(): At time 15.2327s packet sink received 1024 bytes from 10.1.1.30 port 49157 total Rx 2048 bytes
+15.243387203s 6 PacketsSink:HandleRead(): At time 15.2434s packet sink received 1024 bytes from 10.1.1.43 port 49159 total Rx 1024 bytes

```

Slika 1.2: Prikaz terminala

Na slici 1.3 prikazan je *Epidemic routing* protokol u *NetAnim* alatu, gdje se vidi način na koji se vrši kretanje čvorova u mreži.

Slika 1.3: Prikaz kretanja čvorova u *NetAnim* alatu

Ograničenja u realizaciji simulacijskog modela

Ograničenja prilikom modeliranja *Epidemic routing* protokola su nametnuta od strane parametara simulacije. Neki od parametara su ostavljeni na vrijednosti zadanim u skladu sa tabelama 1.1 i 1.2, dok su ostalima mijenjane vrijednosti u skladu sa dokumenom [1].

Jedno od ograničenja predstavlja veličina *buffer*-a. Naime, najbolje performanse se ostvaruju ukoliko čvorovi posjeduju *buffer* sa beskonačnom memorijom. Međutim, zbog nemogućnosti ostvarenja beskonačne memorije performanse ovog protokola će biti umanjene. Ukoliko je veličina *buffer*-a manja od broja paketa dolazi do odbacivanja paketa kada se *buffer* popuni.

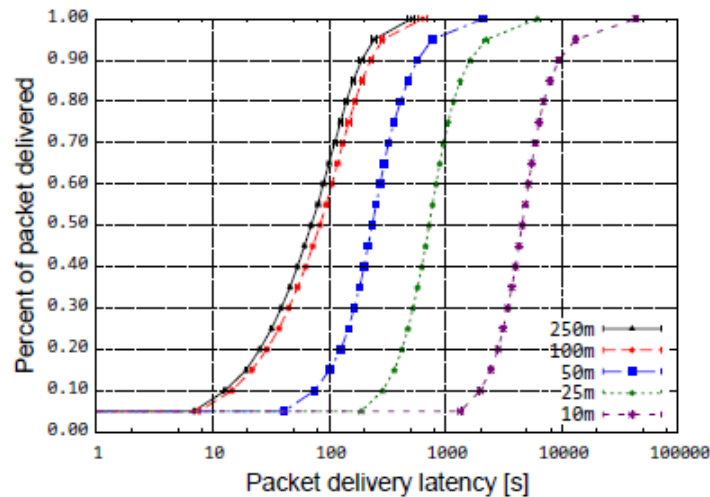
Raspon prenosa između čvorova varira između $10m$ i $250m$ što također predstavlja jedno od ograničenja. Za minimalni raspon od $10m$ kašnjenje, tačnije vrijeme potrebno za isporuku paketa je najveće, dok se povećanjem raspona kašnjenje smanjuje. Tačnije, smanjenjem dometa mreža postaje sve više isprekidana i postoji sve veći broj nepovezanih komponenti koje raposlažu sa malim brojem čvorova, dok povećavanjem raspona transmisije komponente se spajaju zajedno, što kao rezultat daje činjenicu da je kašnjenje manje za veće raspone prenosa.

Ograničenje u realizaciji ovog modela predstavlja i broj *hop*-ova koji utiče na vrijeme kašnjenja. Tačnije, smanjenjem broja *hop*-ova paketi ostaju duže u redovima čekanja dok izvorišni i odredišni čvor ne dođu u domet jedan drugom ukoliko je, na primjer, na raspolaganju samo jedan čvor, čime se povećava vrijeme kašnjenja.

2. Komparacija sa teoretskim rezultatima

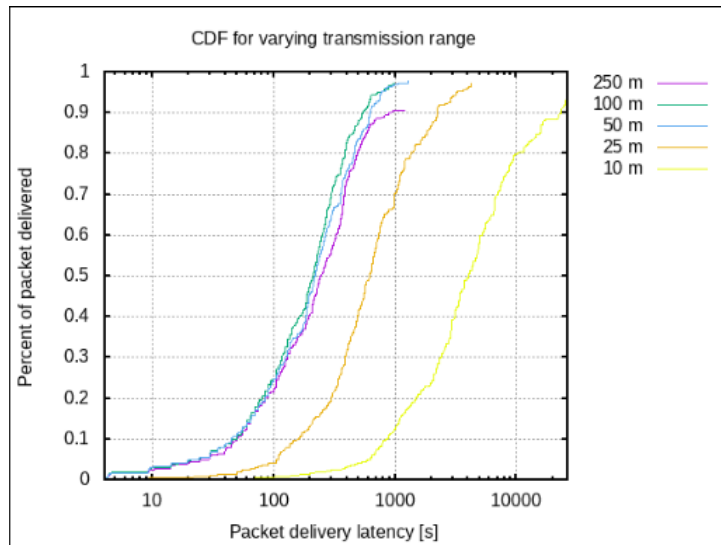
Za ocjenu performansi *Epidemic routing* protokola ispitana su tri scenarija, po uzoru na istraživački rad pod nazivom "*Epidemic Routing Protocol Implementation in ns-3*" [1], koja su međusobno nezavisna. Izvršena je usporedba rezultata dobivenih simulacijom sa teorijskim rezultatima iz navedenog rada. U prvom scenariju se posmatraju performanse protokola u zavisnosti od varijacije veličine raspona prenosa, u drugom scenariju se posmatra kako broj skokova utiče na performanse *Epidemic routing*-a, dok se u trećem scenariju posmatra veličina *buffer*-a i njen uticaj na performanse posmatranog protokola. Svi rezultati su prikazani grafički kroz zavisnost kašnjenja od procenta isporučenih paketa.

2.1. Varijacija veličine raspona prenosa



Slika 2.1: CDF za različite vrijednosti raspona prenosa dobivene teorijski [1]

Sa slike 2.1 možemo vidjeti da povećanje raspona prenosa dovodi do smanjenja kašnjenja isporuke paketa, ali da svi paketi koji su poslani bivaju isporučeni na željeni čvor. Sa druge strane na slici 2.2 koja je dobivena simulacijom, vidimo da rezultati nisu identični i da ima odstupanja u odnosu na teorijski slučaj. Prva stvar koju uočavamo je da procenat uspjehnosti poslanih paketa nije maksimalan ni u jednom slučaju, što znači da u našoj simulaciji dolazi do gubitka paketa. Ovaj gubitak možemo objasniti činjenicom da su čvorovi mobilni i da je u najgorem slučaju svega za 10% poslanih paketa vrijeme koje



Slika 2.2: CDF za različite vrijednosti raspona prenosa dobivene praktično

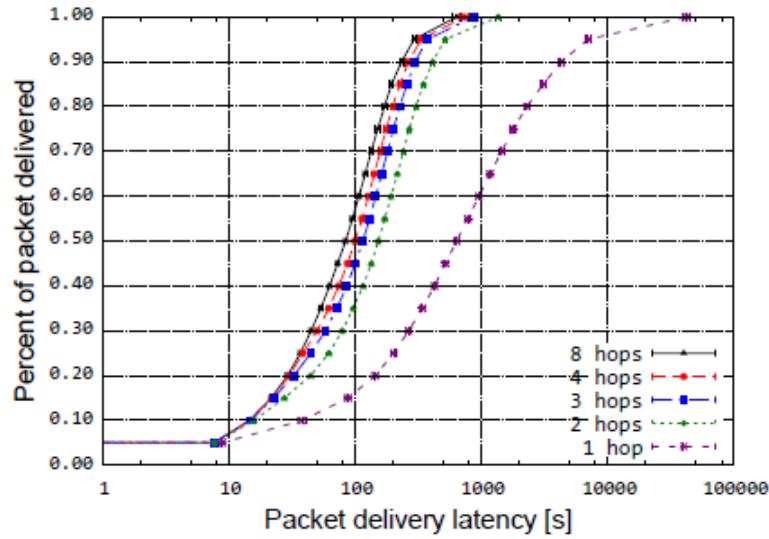
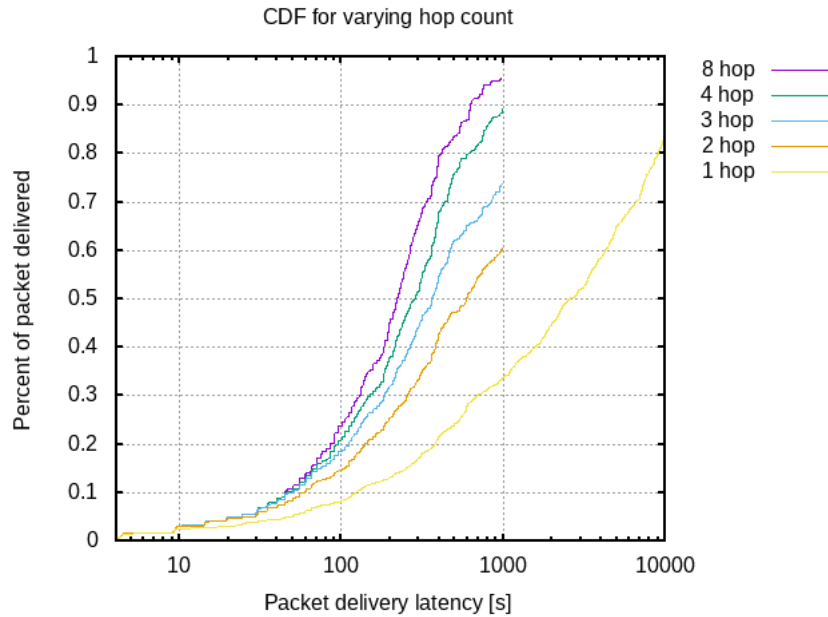
je potrebno da čvor koji šalje paket bude u dometu onih čvora koji ga primaju, veće od simulacijom podešenog vremena koje paket može biti u redu čekanja u nekom od čvorova.

Vidimo da pri malim vrijednostima raspona prijenosa imamo najveća kašnjenja, dok pri većim vrijednostima raspona prenosa, iznad 50m, kašnjenja je približno jednako te je pri rasponu prenosa od 100m kašnjenje najmanje. Možemo zaključiti da, iako simulacijom dobijeni grafik znatno odstupa od teorijskog slučaja u smislu većeg kašnjenja i manjeg procenta uspješno dostavljenih paketa, povećanje kašnjenja za manje vrijednosti raspona prenosa je u skladu sa očekivanjima.

Obzirom da zbog relativno slabe procesorske moći naših računara kao i zbog vremena trajanja jedne simulacije, broj pokretanja modula u NS-3 simulatoru je puno manji u odnosu na onaj koji je predviđen u [1]. Zbog svega navedenog su ova rješenja sasvim realna jer na manjem vremenu ponavljanja puno veći udio uzima vrijeme potrebno da dva čvora budu u dometu jedan drugom i slične pojave koje se dešavaju prilikom simuliranja *Epidemic routing* protokola.

2.2. Varijacija broja hop-ova

Slično kao i u prvom slučaju, različitim varijacijama broja skokova se postiže 100% uspješnost isporuke paketa, i u ovom slučaju je problem kašnjenje isporuke paketa. U slučaju kada je mali broj skokova na raspolaganju paketi duže ostaju u redovima čekanja dok se ne nađe putanja između izvorišnog i odredišnog čvora, to jest dok izvorišni i odredišni čvor ne dođu u domet jedan drugom ukoliko je na raspolaganju samo jedan skok. Dok u slučaju većeg broja skokova ovaj problem nije toliko izražen. Na slikama 2.3 i 2.4 je prikazana normalizovana kumulativna funkcija razdiobe isporuke paketa u zavisnosti od broja *hop*-ova.

Slika 2.3: CDF za varijaciju broja *hop*-ova dobiven teorijski [1]Slika 2.4: CDF za varijaciju broja *hop*-ova dobiven praktično

Sa slike 2.3 vidimo da povećanjem broja *hop*-ova dolazi do smanjenja kašnjenja isporuke paketa, te da je procenat uspješno primljenih paketa u odnosu na broj poslanih paketa maksimalan. Prilikom simuliranja ovakvog scenarija u NS-3 simulatoru dobili smo rezultate koji su prikazani na slici 2.4. Sa slike 2.4 se vidi da se kašnjenje isporuke paketa smanjuje povećanjem broja *hop*-ova, što je u skladu sa očekivanim. Isto tako vidimo da je ukupno kašnjenje za bilo koji odabrani broj *hop*-ova puno veće u simuliranom slučaju nego što je u teorijskom slučaju.

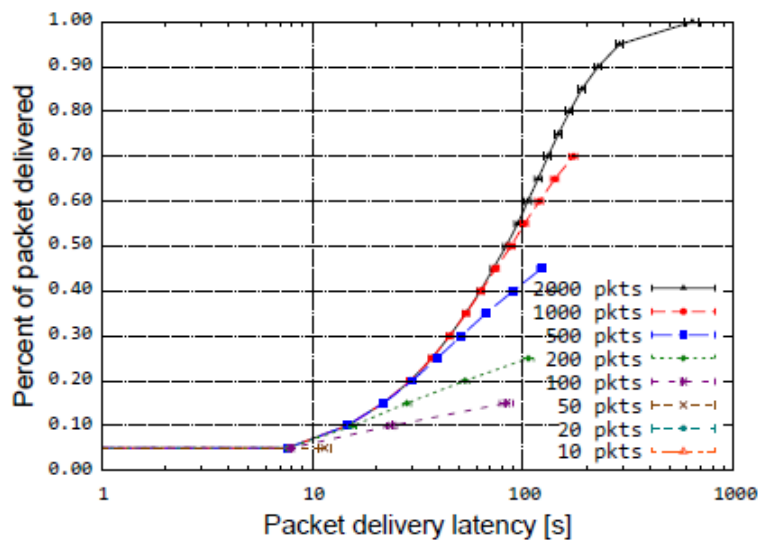
Prilikom simuliranja konkretnog scenarija za sljedeći broj skokova: 8, 4, 3 i 2, zbog ograničenih resursa, odabrano je znatno kraće vrijeme trajanja simulacije (1000 s) jer se, vodeći teorijskim rezultatima sa slike 2.3, očekivalo da će odabrano vrijeme trajanja simulacije biti dovoljno da uspješnost prenosa paketa bude 1 (100 %), u svim slučajevima. Međutim, sa slike 2.4 se uočava da je izvršena pogrešna procjena. Naime, ukupno kašnjenje

je znatno veće u odnosu na isto u teorijskom slučaju, nezavisno o broju *hop*-ova. Iako je ovaj propust doveo do drugačijeg, nepotpunog grafika (slika 2.4), može se uočiti da postoji porast u procentu uspješno dostavljenih paketa pri povećanju broja *hop*-ova. Također se može uočiti da postoji tendencija porasta procenata uspješno dostavljenih paketa za odabir veće vrijednosti trajanja simulacije. Nakon uočavanja navedenog propusta, za broj *hop*-ova koji je jednak 1 je odabrano znatno duže trajanje simulacije, tačnije 10000 s te je pokazano da zaista procenat uspješno dostavljenih paketa teži 1, što se može vidjeti na grafiku.

Ukoliko se ovaj rad nekada u budućnosti bude koristio kao teorijska osnova za simuliranje *Epidemic routing* protokola, potrebno je simulaciju pokretati puno više puta, te povećati vrijeme trajanja simulacije. Ovim izmjenama bi se trebali dobiti rezultati koji su sličniji rezultatima dobivenim u [1], s tim da bi kašnjenje bilo očigledno veće nego u [1].

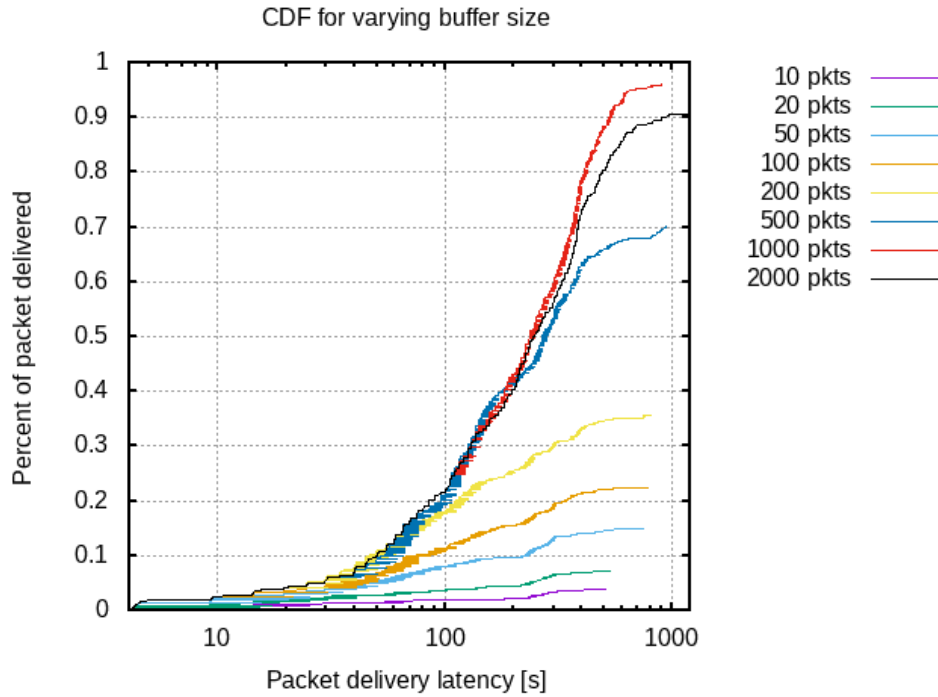
2.3. Varijacija veličine buffer-a

U ovom scenariju je uzeto da veličina *buffer*-a može biti od 10 do 2000 paketa na svakom čvorištu. Vrijednost broja paketa, kao parametra simulacije, je postavljena na 1980 (prikazano u tabeli 1.1) predstavlja ograničavajući faktor na odabir veličine *buffer*-a. Naime, možemo zaključiti da u ovom primjeru vjerovatnoća uspješne isporuke paketa neće biti 100% u svim slučajevima, tačnije u slučajevima gdje je veličina *buffer*-a manja od broja paketa. Oni čvorovi koji ne budu u mogućnosti da odmah procesiraju pakete, nego da ih smjeste u svoj *buffer* mogu dovesti do nepovratnog gubitka paketa ukoliko njihova veličina *buffer*-a nije adekvatno izabrana. Na slikama 2.5 i 2.6 je prikazana normalizovana kumulativna funkcija razdiobe isporuke paketa u zavisnosti od veličine *buffer*-a.



Slika 2.5: CDF za varijaciju veličine *buffer*-a dobiven teorijski [1]

Ako posmatramo sliku 2.5 uočavamo da uspješnost prijema svih poslanih paketa zavisi od veličine *buffer*-a čvorova koji se nalaze u mreži. Jedino u slučaju kada u *buffer* jednog čvora mogu stati svi paketi (1980 njih koji se šalju u simulaciji), neće doći do gubitka

Slika 2.6: CDF za varijaciju veličine *buffer*-a dobiven praktično

paketa zbog popunjenosti *buffer*-a. Prilikom simuliranja ovog scenarija dobiveni su rezultati koji su prikazani na slici 2.6. Sa slike se vidi da prilikom porasta veličine *buffer*-a raste i procenat uspješno primljenih paketa, čak je u slučaju naše simulacije taj procenat veći nego što je slučaj u [1]. Također, kao i u prethodnim scenarijima testiranja, ukupno kašnjenje za bilo koju odabranu veličinu *buffer*-a je veće u simuliranom slučaju, što se uočava poređenjem grafika na slikama 2.5 i 2.6.

Za vrijednost trajanja simulacije koju smo uzeli kao parametar iz posmatranog rada dobivamo da je procenat uspješnosti paketa za veličinu *buffer*-a 1000 veći nego za veličinu *buffer*-a 2000, što svakako nije u skladu sa očekivanjem. Međutim, povećanjem broja pokretanja simulacija i vremena trajanja simulacije očekuje se porast u procentu uspješno dostavljenih paketa za veličinu *buffer*-a koja iznosi 2000.

3. Diskusija i zaključak

Cilj ovog rada je bio prije svega analizirati rješenje, tačnije kod u NS-3 simulatoru, te nakon toga simulirati *Epidemic routing* protokol i uporediti dobivena rješenja sa rješenjima datim u [1]. *Epidemic routing* protokol prenosi pakete principom 'zaražavanja' susjednih mobilnih čvorova, isti nastoji osigurati što je moguće manje vrijeme kašnjenja uz velike brzine i veliku vjerovatnoću uspješnog prenosa. *Epidemic routing* protokol je izuzetno pogodan kod rijetkih ili isprekidanih mreža, a najbolje performanse ostvaruje za mrežu u kojoj čvorovi posjeduju *buffer* sa beskonačnom memorijom.

Nakon odrađenih simulacija i crtanja grafika za *Epidemic routing* protokol, u ovom radu izvršena je komparacija dobivenih rješenja sa teorijskim rješenjima datim u dokumentu [1]. Izvršeno je poređenje različitih faktora koji utiču na sam protokol, te su date neke sugestije kako je moguće poboljšati performanse samog protokola. Odnosno koje izmjene je potrebno napraviti kako bi se poboljšao prenos u smislu kašnjenja, propusnosti itd.

Prije svega izvršena je komparacija dobivenih rezultata sa teorijskim, tj. za rezultatima navedenim u [1] kada je u pitanju veličina raspona prenosa, broj *hop*-ova i veličina *buffer*-a. Kada je u pitanju veličina raspona prenosa, povećanjem raspona prenosa dolazi do smanjenja kašnjenja, također simulacijsko rješenje nema maksimalan procenat uspješnosti poslanih paketa, štp znači da dolazi do gubitaka paketa. Do smanjenja kašnjenja isporuke paketa također dolazi i usljed povećanja broja *hop*-ova. Ukupno kašnjenje je veće u simuliranom slučaju u odnosu na teorijski slučaj, za bilo koji odabrani broj *hop*-ova. Također, uspješnost prijema svih poslanih paketa zavisi od veličine *buffer*-a čvorova koji se nalaze u mreži. Kako se povećava veličina *buffer*-a određenog paketa, kašnjenje se smanjuje, međutim do gubitaka ne dolazi jedino u slučaju kada u *buffer* određenog čvora mogu stati svi paketi koji se šalju kroz mrežu. Svi dobijeni rezultati kao i njihova komparacija sa teorijskim su prethodno detaljno objašnjeni.

Treba primijetiti da na simulaciju *Epidemic routing* protokola u NS-3 simulatoru, pored navedenih zantno utiču i drugi parametri. Pod ovim parametrima podrazumijevaju se procesorska moć računara na kojima se vrši simuliranje, kao i vremenski rok za izradu, odnosno simuliranje datog protokola.

Iako rezultati dobiveni simulacijom odstupaju od očekivanih rezultata iz [1] u pogledu kašnjenja (koje je u svim slučajevima veće od očekivanih vrijednosti) i procenta uspješno dostavljenih paketa (koje je u većini slučajeva manje od očekivanih vrijednosti), možemo zaključiti da je uspješno izvršena simulacija *Epidemic routing* protokola. U budućim simuliranjima *Epidemic routing* protokolima trebalo bi se uzeti duže vrijeme trajanja simulacije, izvršiti pokretanje simulacije više puta, te koristiti paralelne procese radi dobijanja vjerodostojnijih rezultata.

Popis slika

1.1	Prikaz terminala	3
1.2	Prikaz terminala	4
1.3	Prikaz kretanja čvorova u <i>NetAnim</i> alatu	4
2.1	CDF za različite vrijednosti raspona prenosa dobivene teorijski [1]	6
2.2	CDF za različite vrijednosti raspona prenosa dobivene praktično	7
2.3	CDF za varijaciju broja <i>hop</i> -ova dobiven teorijski [1]	8
2.4	CDF za varijaciju broja <i>hop</i> -ova dobiven praktično	8
2.5	CDF za varijaciju veličine <i>buffer</i> -a dobiven teorijski [1]	9
2.6	CDF za varijaciju veličine <i>buffer</i> -a dobiven praktično	10

Popis tablica

1.1	Parametri simulacije [1]	2
1.2	Parametri evaluacije <i>Epidemic Routing</i> protokola [1]	3

Bibliografija

- [1] M. J. F. Alenazi, Y. Cheng, D. Zhang, and J. P. G. Sterbenz, "Epidemic routing protocol implementation in ns-3," in *WNS3*, 2015.
- [2] A. Kojić, "Simulacija procesa u telekomunikacijskim mrežama - Laboratorijska vježba 4," 2019/2020.
- [3] "Epidemic Routing - Epidemic Routing for Partially-Connected Ad Hoc Networks." [Online]. Available: <https://apps.nsnam.org/app/epidemic-routing/>
- [4] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," 2000.
- [5] L. Wan, F. Liu, Y. Chen, and H. Zhang, "Routing Protocols for Delay Tolerant Networks: Survey and Performance Evaluation," 2015.
- [6] X. Zhang, G. Neglia, J. F. Kurose, and D. F. Towsley, "Performance Modeling of Epidemic Routing," in *Networking*, 2006.
- [7] Y. Lin, B. Li, and B. Liang, "Stochastic analysis of network coding in epidemic routing," *IEEE Journal on Selected Areas in Communications*, vol. 26, 2008.
- [8] R. Thakur and K. L. Bansal, "Delay Tolerant Networks: An Analysis of Routing Protocols with ONE Simulator," 2016.
- [9] F. Herbertsson, "Implementation of a Delay-Tolerant Routing Protocol in the Network Simulator NS-3," 2010.
- [10] J. P. Rohrer and A. N. Mauldin, "Implementation of Epidemic Routing with IP Convergence Layer in ns-3," in *WNS3*, 2018.
- [11] "InterPlanetary Networking Special Interest Group (IPNSIG)." [Online]. Available: <http://ipnsig.org/introducing-delay-disruption-tolerant-networking-dtn/>
- [12] M. Aneja and V. Garg, "Simulation of Epidemic, Spray and Wait and First Contact Routing Protocols in Delay Tolerant Network," *IOSR journal of electronics and communication engineering*, pp. 30–34, 2015.
- [13] F. De Rango, S. Amelio, and P. Fazio, "Enhancements of epidemic routing in delay tolerant networks from an energy perspective," in *2013 9th international wireless communications and mobile computing conference (IWCMC)*. IEEE, 2013, pp. 731–735.
- [14] B. B. Bista and D. B. Rawat, "EA-Epidemic: An Energy Aware Epidemic-Based Routing Protocol for Delay Tolerant Networks," *JCM*, vol. 12, pp. 304–311, 2017.
- [15] Z. Feng and K.-W. Chin, "A Unified Study of Epidemic Routing Protocols and their

- Enhancements,” *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum*, pp. 1484–1493, 2012.
- [16] S. Gupta, R. C. Poonia, and X.-Z. Gao, “Performance Evaluation of Epidemic Routing Protocol for Delay Tolerant Network Based on Node Movement Model.”
 - [17] L.-Å. Norden, “Simulation and Analysis of Opportunistic Routing Protocols,” 2013.
 - [18] T. Henderson, “Epidemic Routing,” 2017. [Online]. Available: <https://github.com/tomhenderson/epidemic-routing/blob/master/doc/epidemic.rst>
 - [19] W. Navidi, T. Camp, and N. B. Bauer, “Improving the Accuracy of Random Waypoint Simulations Through Steady-State Initialization,” 2003.
 - [20] A. N. Mauldin, “Comparative Analysis of Disruption Tolerant Network Routing Simulations in the One and NS-3,” 2017.
 - [21] Q. Wang and Z. J. Haas, “Analytical Model of Epidemic Routing for Delay-Tolerant Networks,” in *Proceedings of the 1st ACM workshop on High performance mobile opportunistic systems*. ACM, 2012, pp. 1–8.
 - [22] Itf, “Shared Mobility Simulations for Helsinki,” 2017.