

Софтвер за анализу перформанси WMN у реалном окружењу, заснован на мрежном симулатору GloMoSim

Руководилац пројекта: Мирјана Стојановић

Одговорно лице: Марија Малнар

Аутори: Марија Малнар

Развијено: у оквиру пројекта технолошког развоја TP-32025

Година: 2013.

Примена: 01.10. 2012.

Кратак опис

Развијен је скуп програма на програмском језику Parsec за симулацију бежичних mesh мрежа помоћу мрежног симулатора Glomosim. Parsec је симулациони језик заснован на C програмском језику који је развијен од стране Parallel Computing лабораторије на UCLA Универзитету за секвенцијално и паралелно извршавањем симулационих модела са дискретним догађајима. Он се такође може користити као програмски језик за паралелно извршавање. У симулатор је имплементиран LQSR (*Link Quality Source Routing*) протокол рутирања као и три метрике рутирања које овај протокол може да користи: ETX (*Expected Transmission Count*), ETT (*Expected Transmission Time*) и MIC (*Metric of Interference and Channel Switching*). Симулирана је мрежа по угледу на реалну мрежу реализовану на Електротехничком факултету у Београду. Извршен је развој програма за жељени сценарио симулације. Избором одговарајућих улазних параметара симулације могуће је извршити низ анализа за различите метрике, мреже са различитим бројем чворова и различита саобраћајна оптерећења.

Техничке карактеристике:

Мрежни симулатор GloMoSim, Windows платформа, програмски језик Parsec, протокол рутирања LQSR, метрике рутирања ETX, ETT и MIC.

Техничке могућности:

Софтвер отвореног типа, погодан за примену у научно-истраживачке и едукативне сврхе.

Реализатори:

СФ Београд, ЕТФ Београд

Корисници:

СФ Београд, ЕТФ Београд

Подтип решења:

Софтвер (M85)

Стање у свету

Мобилна mesh мрежа (*Wireless Mesh Network*, WMN) представља комуникациону мрежу која се састоји из радио чворова организованих у mesh топологију. Под појмом mesh топологија подразумева се мрежна топологија у којој сваки чвор није задужен само за своје податке већ служи и као релеј за остале чворове, и преноси податке ка другим чворовима у мрежи.

WMN се састоји из mesh клијената, mesh рутера и gateway чворова. Mesh клијенти су углавном лаптоп рачунари, мобилни телефони и остали бежични уређаји, док су mesh рутери уређаји који служе са усмеравање саобраћаја од и ка

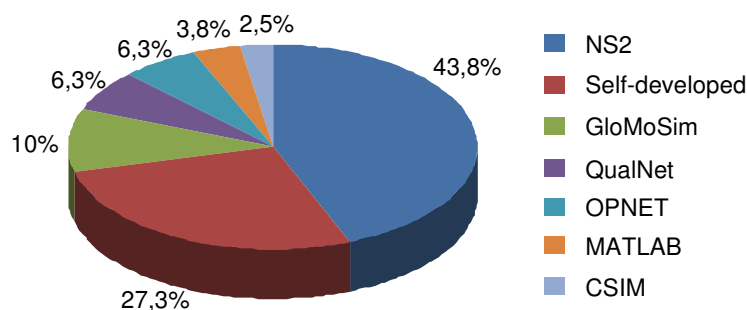
gateway чворовима. Преко *gateway* чворова мрежа се повезује са интернетом или неком другом мрежом.

С обзиром да WMN мреже нису довољно распрострањене истраживања у овој области углавном се одвијају на основу симулација мреже. У последње време алати за симулацију се константно унапређују и проширују новим протоколима и карактеристикама. Сви симулатори бежичних мрежа морају обезбедити скалабилност, ефикасно извршавање симулације и уштеду рачунарских ресурса. У Табели 1 дат је приказ најзаступљенијих симулатора који су тренутно доступни на тржишту. Симулатори се разликују по цени, једноставности употребе, доступности кода, обиму документације исл.

Табела 1. Преглед најзаступљенијих симулатора

Алат за симулацију и моделовање мрежа	Доступност	URL
BRITE	<i>Open-source</i>	http://www.cs.bu.edu/brite/
Cnet	<i>Open-source</i>	www.csse.uwa.edu.au/cnet/
GloMoSim	<i>Open-source</i>	http://pcl.cs.ucla.edu/projects/glomosim/
J-Sim	<i>Open-source</i>	http://sites.google.com/site/jsimofficial/
NS2	<i>Open-source</i>	http://www.isi.edu/nsnam/ns/
OMNeT++	<i>Open-source</i>	http://www.omnetpp.org/
OPNET	Комерцијалан	http://www.opnet.com/
PacketStorm-Network Emulator	Комерцијалан	http://www.packetstorm.com/4xg.php
Qualnet	Комерцијалан	http://www.scalable-networks.com/
SSFNet	<i>Open-source</i>	http://www.ssfnet.org/homePage.html
X-sim	<i>Open-source</i>	http://www.cs.arizona.edu/projects/xkernel/

Студија која је обухватила више од 150 научно-истраживачких радова из области мобилних *ad hoc* мрежа (MANET) статистички је представила заступљеност одређених симулатора. Подаци су показали да је већина аутора користила симулатор NS2, као што је приказано на Слици 1. После NS2 симулатора, велики број аутора користио је симулаторе које су сами развили, а први следећи симулатор који се користи је управо GloMoSim.



Слика 1. Примењени симулатори у области истраживања MANET мрежа.

Структура симулатора GLOMoSIM

GloMoSim (*Global Mobile System Simulator*) је скалабилно симулационо окружење за мобилне и бежичне мреже, развијен је у *Parallel Computing Laboratory* на UCLA универзитету у Лос Анђелесу, Калифорнија, Сједињене Америчке државе. Он је симулатор дискретних догађаја развијен коришћењем PARSEC (*for PARAllel Simulation Environment for Complex systems*) програмског језика.

PARSEC је програмско окружење засновано на C програмском језику дизајнирано за паралелне симулације. Развијено је у *Parallel Computing Laboratory* на UCLA универзитету. Може се инсталирати на већини актуелних UNIX оперативних система као и у Windows окружењу.

PARSEC се заснива на приступу симулацији са дискретним догађајима. Под дискретним догађајима подразумева се извршавање програма које се састоји од скупа догађаја (*event*). Када се догађај деси, као одговор на догађај изврши се одговарајућа акција која мења стање система. Неки догађај, или комбинација догађаја за последицу имају неке друге догађаје и на тај начин се извршавају процеси. Пример догађаја је тренутак када стигне пакет у неки одређени чвор, или када истекне неки дефинисани тајмера у симулатору. Догађаји се остварују у дискретним јединицама времена и не могу се десити између тих дефинисаних временских тренутака. Интеракција догађаја моделована је разменом порука између одговарајућих логичких слојева које се дешавају у неком тренутку дефинисаном у симулацији.

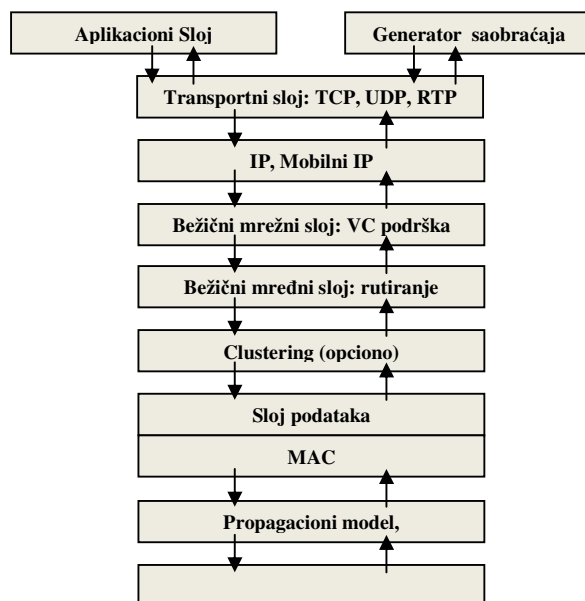
Архитектура GLOMoSIM

Као и већина мрежних система и GloMoSim је дизајниран да користи приступ заснован на слојевима сличним OSI мрежној архитектури са стандардним API (*Access Point Interface*) између њих, као што је приказано на Слици 2. Једноставан чвор се дефинише између различитих симулационих слојева. Дизајн са слојевима приступачан је јер ствара модуларно окружење. Под тим се подразумева да се протоколи и модели на различитим слојевима третирају независно и могу се моделовати или заменити без утицаја на протоколе и моделе на другим слојевима. Модулациони дизајн такође омогућава корисницима да развијају и имплементирају нове протоколе на различитим слојевима тако да дизајн тих протокола одговара стандардним API између слојева. GloMoSim је скалабилни симулатор па има могућност симулација са хиљадама чворова.

Када се инсталира симулатор добија се у фолдеру /glomosim следећи подфолдери:

- /application – у ком се налазе кодови за апликациони слој
- /bin – и ком се налазе егзикуциони и улазно/излазни фајлови
- /doc – у ком се налази документација
- /include – у ком се налазе заједнички include документи
- /java gui – у ком се налази tool за визуелизацију
- /mac – у ком се налазе кодови за mac слој
- /main - садржи основне *framework* дизајн програме
- /network – у ком се налазе кодови за мрежни слој
- /radio – у ком се налазе кодови за радио слој

- /scenarios – у ком се налазе примери неких сценарија
- /tcplib – у ком се налазе библиотеке за TCP
- /transport – у ком се налазе кодови за транспортни слој



Слика 2. Архитектура GloMoSim симулатора.

У фолдеру за сваки слој постоје одговарајући програми који позивају за одговарајући протокол. Тако постоје програми `radio.pc`, `propagation.pc`, `mac.pc`, `network.pc`, `transport.pc` и `application.pc`. У фолдеру `/include` налазе се заглавља одговарајућих програма који се позивају.

У фолдеру `/main` налазе се основни програми који служе за извршавање GloMoSim програма. Најважнији од њих су: `driver.pc`, `glomo.pc`, `message.pc` и `nodes.pc`.

У `driver.pc` фајлу дефинише се ентитет `driver`. Сваки Parsec програм мора да садржи ентитет `driver` који има улогу функције `main` у програмском језику C. Овај фајл чита улазни фајл (`config.in`), иницира симулацију и испишује коначне статистичке резултате (у фајл `glomo.stat`). Секвенца догађаја који се покрећу је:

1. Прво се покреће `main` функција у `driver.pc` фајлу. То је `main` функција програмског језика C и ту почиње GloMoSim симулација.
2. `Main` функција у `driver.pc` позива `parsec_main()` и покреће Parsec симулациони програм, иницијализује временске променљиве у симулатору и креира ентитет `driver`. Функција `parsec_main()` користи се и када корисник жели да напише своју `main` функцију. Налази се у директоријуму `PCC DIRECTORY/include/pc_api.h`.
3. Када се симулација заврши враћа се на место где је позвана `parsec_main()` функција и извршава се остатак `main` функције.

Ентитет `driver` који је креиран у `driver.pc` фајлу, чита улазне податке, дефинише партиције, алоцира меморију за информације о чворовима, позива одговарајуће функције у зависности од учитаних вредности као што су симулационо време и

позиција чворова, и почиње симулацију слањем *StartSim* поруке ентитету *partitionEntityName* (који је ентитет типа GLOMOPartition дефинисана у *gloмо.рс* фајлу). Дакле сам симулатор представља комуникацију између два ентитета *driver* и GLOMOPartition, симулација почиње када *driver* пошаље поруку *StartSim* ентитету GLOMOPartition.

Након што ентитет GLOMOPartition прими поруку *StartSim* од ентитета *driver* симулација почиње са извршавањем. Ентитет GLOMOPartition налази се у другом важном фајлу у */main* директоријуму, *gloмо.рс* Он извршава следеће кораке:

1. Прима информације од *driver* ентитета.
2. Проналази чворове који припадају тренутној партицији и иницијализује све слојеве за те чворове позивањем следећих ентитета GLOMO PropInit(), GLOMO RadioInit(), GLOMO MacInit(), GLOMO NetworkInit(), GLOMO TransportInit(), GLOMO AppInit() and GLOMO MobilityInit().
3. Одлази у петљу у којој покушава да прими поруку. Када је порука примљена он враћа информације о чвору који је уписан у поруци и позива одговарају функцију (која се такође налази у поруци). Такође приказује тренутно симулационо време у зависности од извршавања програма.
4. Када се заврши симулација, он одлази до функције *Finalize* за све слојеве и све чворове у партицији. На тај начин могу се покупити све потребне статистике. Функције за финализацију су: GLOMO RadioFinalize(), GLOMO MacFinalize(), GLOMO NetworkFinalize(), GLOMO TransportFinalize(), GLOMO AppFinalize() and GLOMO MobilityFinalize().

Споменуте функције служе са иницијализацију, позивање и финализацију свих чворова и дефинисане су у *./Include/structmsg.h* и кодиране у одговарајућем фолдеру. На пример GLOMO RadioInit() је кодиран у *./Radio/radio.рс*, док је GLOMO MacInit() кодиран у *./mac/mac.рс*, GLOMO NetworkInit() у *./network/network.рс*, GLOMO TransportInit() у *./transport/transport.рс*, итд.

Дизајн GLOMoSIM

Као што је већ речено GloMoSim је дизајниран по принципу слојева са стандардним API између два различита слоја. Да би се једноставно имплементирао нови протокол, сваки слој је представљен као посебан ентитет. На тај начин укључени су модели за канал, радио, MAC, мрежни, транспортни и више слојеве. Модели и протоколи који су подржани на различитим слојевима приказани су у Табели 2.

У оквиру сваког од слојева имплементирано је више различитих протокола: за апликациони слој Telnet, FTP, HTTP и CBR, за транспортни UDP, TCP и RTP, *unicast* и *multicast* протоколи рутирања Bellman-Ford, FSR, OSPF, DSR, WRP, LAR и AODV, MAC протоколи CSMA, MACA, TSMA, и 802.11 као и две врсте радио пропагације *Free space*, *Two-Ray*. У оквиру радио слоја имплементиран је радио канал са и без шума као и две врсте фединга: фединг за Rayleigh расподелом и фединг са Rician расподелом.

Табела 2. Протоколи и модели који су подржани на сваком слоју.

Слој	Протокол
Мобилност	Random Waypoint, Trace Based
Радио пропација	Two ray, Free space
Радио модел	Noise Accumulating
Модел примања пакета	SNR, BER са BPSK/QPSK модулацијом
MAC	CSMA, IEEE 802.11, MACA
Мрежни (Рутирање)	IP: AODV, Bellman-Ford, DSR, Fisheye, LAR, ODMRP, WRP
Транспортни	TCP, UDP
Апликативни	CBR, FTP, HTTP, Telnet

GloMoSim подржава мобилност чворова. Чворови се могу кретати по моделу *Random Waypoint Mobility Model (RWPM)*. У овом моделу чвор случајно бира дестинацију на основу описа терена и креће се у правцу ка тој дестинацији брзином која је униформно изабрана између вредности MOBILITY-WP-MIN-SPEED и MOBILITY-WP-MAX-SPEED (које чита из config.in фајла) Након што чвор стигне до те дестинације, у њој остаје временски период MOBILITY-WP-PAUSE. Ако је потребно користити други тип мобилности осим RWPM потребно је у config.in фајлу дефинисати параметар MOBILITY_TRACE и специфицирати у посебном фајлу MOBILITY-TRACE-FILE трасу којом ће се кретати чвор. Параметар MOBILITY-INTERVAL користи се да на сваки MOBILITY-INTERVAL период иницира чвору да освежи свој податак о позицији, док се MOBILITY-D-UPDATE параметар користи када чвор учита свој положај на основу растојања (у метрима). Сви параметри мобилности дефинисани су у config.in фајлу.

GloMoSim има неколико уграђених тајмера на различитим слојевима. Уопштено помоћу тајмера се остварују догађаји који се дешавају за одређени чвор. Фајл */glomosim/include/structmsg.h* дефинише догађаје у GloMoSim, а корисник може да дефинише своје догађаје и тајмере.

Пакети и поруке који пролазе кроз слојеве морају у себи имати информације о томе са ког слоја су послати и ком слоју се шаљу. Слање пакета и порука остварује се заказивањем догађаја (*Scheduling Events*) и сваком догађају мора бити додељен његов јединствени “*Event type ID*”. У симулатору постоје две врсте порука:

- Пакети (*Packets/ Cell Messages*): пакети који се размењују између чворова (*Inter-node packets*) или пакети који се размењују између слојева (*Inter-layer packets*).
- Поруке које не садрже мрежне пакете (*Non-Packet Messages*): поруке које описују догађаје између слојева (*Inter-layer event messages*) и догађаји који представљају само постављене тајмере (*Self scheduled (timer) events*).

Под *Packets/Cell Messages* подразумевају се пакети који би се слали у стварној мрежи. *Inter-layer* пакети су пакети који се размењују између слојева сваког од чворова, примера ради када мрежни слој проследи пакет на MAC слој, па MAC слој дода своје заглавље пакету и проследи га радио слоју. *Inter-node* пакети

представљају стварне пакете који би се у мрежи преносили између два чвора, и који се емитују на физичком слоју.

С друге стране, да би сама симулација могла да се оствари постоје и поруке које у реалној мрежи не постоје тј поруке које не садрже никакве пакете већ служе за комуникацију самог симулатора и извршавање догађаја. То су *Non-Packet Messages*. Те поруке могу бити поруке које описују догађаје који треба да се догоде између два слоја (*Inter-layer event messages*) или поруке везане за тајмере (*Self scheduled (timer) events*).

Да би се овај процес једноставније објаснио, нека је потребно да у DSR протоколу рутирања чвор жели да пошаље *Route Request* (RREQ) пакет. Тада ће чвор позвати функцију *RoutingDsrInitiateRREQ* (чији се код и дефиниција налазе у фајловима */network/dsr.pc* и */network/dsr.h*). Последњи ред ове функције позива команду:

```
RoutingDsrSetTimer(node, MSG_NETWORK_CheckReplied,  
                  cr, (clocktype)DSR_RING0_REQUEST_TO);
```

Функција *RoutingDsrSetTimer* поставља тајмер тако што шаље поруку *MSG_NETWORK_CheckReplied* одговарајућем мрежном слоју. Последња променљива у позивању функције је *DSR_RING0_REQUEST_TO* је кашњење, тј. симулационо време након ког ће се послати ова порука, и дефинисано је на 30ms у *dsr.h* фајлу.

Функција *RoutingDsrSetTimer* у себи садржи функције *GLOMO_MsgAlloc*. Том функцијом се заузима меморија за променљиву *newMsg* у коју се уписује врста догађаја који треба да се деси (*MSG_NETWORK_CheckReplied*). Помоћу параметра *GLOMO_NETWORK_LAYER* дефинише се на ком слоју је генерисана порука, а *ROUTING_PROTOCOL_DSR* дефинише који протокол је генерисао поруку. На крају се позива функцију *GLOMO_MsgSend* којом се порука *newMsg* шаље након времена *delay*. Када исткне тајмер (након симулационог времена *delay*) позива се функција *RoutingDsrHandleProtocolEvent*. У њој се проналази део кода за догађај *MSG_NETWORK_CheckReplied* којим се проверава да ли је откривена путања након истека 30 ms од слања RREQ пакета. Ако је откривена путања, порука се ослобађа, а ако није покреће се поновно слање RREQ пакета.

Додавање нових компоненти

Главна предност свих слободно доступних симулатора јесте могућност додавања нових или измена постојећих протокола и алгоритама за потребе. Основни недостатак GloMoSim симулатора у односу на комерцијалне симулаторе, као на пример QualNet (који представља комерцијалну верзију GloMoSim симулатора) су недостатак литературе за коришћење симулатора, лош графички интерфејс и немогућност квалитетне визуелизације симулације. Предност је то што је симулатор писан у Parsec симулационом језику који је врло сличан C програмском језику, који је један од основних програмских језика, и широко је распрострањен. Међутим само компајлирање и повезивање нових или измене постојећих протокола захтевају одлично познавање не само материје и протокола већ и програмирања у

симулационим пакетима. Још један од недостатака је то што је симулатор развијен 1999. године након чега га је преузела фирма ScalableNetworks и развијла комерцијалну верзију QualNet, па сам GloMoSim симулатор нема развијене најновије стандарде и протоколе.

Додавање нових компонената симулатору (нпр. нова или модификована верзија неког протокола, нови стандарди исл.) врши се писањем новог програма екстензије .pc за одређени стандард (нпр додавање новог протокола рутирања). Након допуне изворног кода, програм се памти у главном програму за одговарајући слој у случају протокола рутирања у фолдер мрежног слоја). За сваки слој постоје одговарајући програми (radio.pc, propagation.pc, MAC.pc, network.pc, transport.pc и application.pc) који позивају одговарајући протокол. Када се програм повеже са постојећим језгром симулатора, врши се компилација и линковање са постојећим библиотекама у batch фајлу makent.bat који формира ехе верзију симулатора, glomo.exe који се користи за даље пуштање симулације.

Имплементација протокола рутирања

За потребе анализе бежичних *mesh* мрежа у симулатор је имплементиран нови протокол рутирања LQSR (*Link quality source routing*) који представља модификацију DSR (*dynamic source routing*) протокола. Имплементација новог протокола рутирања подразумева да се у потпуности дефинишу сви параметри карактеристични за одговарајуће симулирано окружење и примену симулиране мреже.

LQSR протокол рутирања задржава све особине и механизме DSR. Модификација протокола огледа се у томе што се поред адресе чворова кроз које пакети пролазе, преносе и цене линкова између два чвора која су на тој путањи. Два основна механизма код DSR и LQSR протокола су *Route Discovery* и *Route Maintenance*.

Помоћу *Route Discovery* механизма изворишни чвор S, који жели да пошаље пакет одредишном чвору D, проналази путању којом ће слати пакете. Овај механизам остварује се помоћу две врсте пробних пакета, RREQ (*Route Request*) и RREP (*Route Replay*). Код LQSR протокола одредишни чвор осим копије путање у *Route Replay* уписује и одговарајуће тежине линкова који одговарају путањи у пристиглом *Route Request* пакету. Када се примењује DSR протокол он бира путање са најмањим бројем чворова, а када се примењује LQSR протокол, он бира путању са најмањим збиром тежина линкова.

Route Maintenance је механизам којим изворишни чвор S, који већ зна путању до одредишног чвора D, коме жели да пошаље пакет, проверава да ли путања идалје може да се користи за слање података, или је неки линк на тој путањи у прекиду. Ако је неки линк на путањи у прекиду генерише се REER (*Route Error*) пакет којим се обавештава чвор S да та путања није више за употребу.

LQSR протокол је имплементиран у симулатор у посебним фајловима lqsr.pc и lqsr.h и повезан са симулатором преко nwip.pc фајла на мрежни слој. Да би се користио нови протокол неопходно је имплементирати и неку од постојећих метрика рутирања како би се рачунале цене путања.

Имплементиране метрике

За потребе анализе LQSR протокола, у симулатор су имплементиране три метрике: ETX (*Expected transmission count*), ETT (*expected transmission time*) и MIC (*Metric of Interference and Channel-switching*).

ETX проналази путање са најмањим бројем ретрансмисија пакета потребних да се пакет успешно пошаље од изворног до одредишног чвора. Број ретрансмисија пакета дуж линка одређује се мерењем вероватноће успешног преноса пакета. Тежина линка израчунава се као:

$$ETX = \frac{1}{p_f \cdot p_r} \quad (1)$$

Где је p_f вероватноћа да пакет успешно стигне до чвора коме је послат, а p_r вероватноћа да АСК пакет успешно стигне до чвора који је иницирао слање.

ETT метрика је модификација ETX метрике која у обзир узима и различите капацитете линкова. ETT представља очекивано време потребно да се пакет успешно пренесе на MAC слоју. За линк са тежином ETX, ETT се рачуна као $ETT = ETX \cdot S/B$, где је S просечна величина пакета, а B максимални проток који може да се оствари тим линком.

MIC је такође модификација ETX метрике, и састоји се из две компоненте: IRU (*Interference aware Resource Usage*) која урачунава у тежину линка утицај *inter-flow* интерференције и CSC (*Channel Switching Cost*) којом се у цену линка урачунава утицај *intra-flow* интерференције.

IRU компонента рачуна укупно очекивано време преноса на неком каналу c , између чворова i и j , узимајући у обзир утицај оних суседа чворова i и j , који стварају интерференцију сигналу који се преноси на каналу c .

$$IRU_{ij}(c) = ETT_{ij}(c) \times |N_i(c) \cup N_j(c)| \quad (2)$$

Где је $N_i(c)$ скуп суседа чвора i који стварају интерференцију када чвор i шаље податке преко канала c . $|N_i(c) \cup N_j(c)|$ је укупан број суседних чворова који могу да стварју интерференцију приликом слања података између чворова i и j преко канала c .

CSC компонента узима у обзир утицај *intra-flow* интерференције. Нека је $prev(i)$ претходни чвор чвора i , а $CH(i)$ канал који чвор i користи за пренос података до следећег чвора. Тада је CSC чвора i :

$$CSC_i = \begin{cases} \omega_1, & CH(prev(i)) \neq CH(i) \\ \omega_2, & CH(prev(i)) = CH(i) \end{cases}, \quad 0 \leq \omega_1 < \omega_2 \quad (3)$$

Веза $\omega_1 < \omega_2$ треба да послужи да у случају да чвор i емитује податке на истом каналу по ком су му подаци стигли, он повећава *intra-flow* интерференцију у односу на случај када за предају и пријем користи различите канале. Комбинацијом IRU и CSC дефинише се MIC за путању p као:

$$MIC(p) = \frac{1}{N_n \times \min ETT} \sum_l IRU_l + \sum_i CSC_i \quad (4)$$

Где је N_n број чворова у мрежи, $\min ETT$ најмањи ЕТТ у мрежи, прва сума је сума по l линковима који се налазе на путањи p , а друга сума је по i чворова који се налазе на путањи p .

Пример коришћења софтвера

Извршавање програма у симулатору састоји се из неколико компоненти.). Након успешног инсталирања симулатора, програм се извршава из MsDos-а куцањем следеће команде:

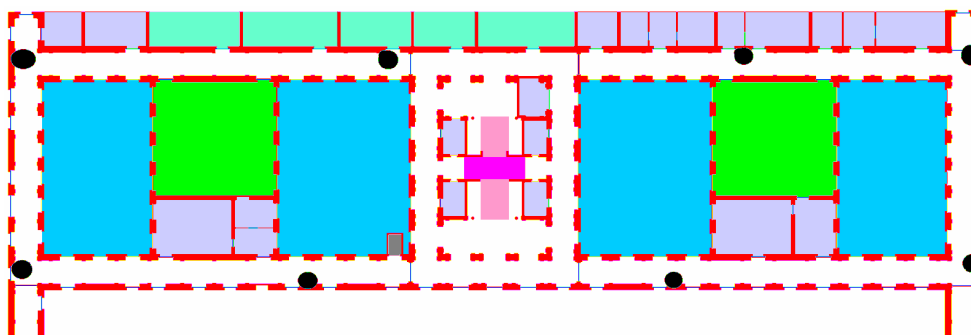
. /glomosim < inputfile >

<inputfile> је у ствари основни конфигурациони улазни фајл у ком су уписани параметри за симулацију (*config.in*).

Други неопходан улазни фајл је и *app.conf* у ком се дефинише врста саобраћаја која се остварује као што је FTP или Telnet. Тренутно расположиви генератори саобраћаја су FTP, FTP/GENERIC, TELNET, CBR, и HTTP.

Анализа резултата симулације обавља се на основу *glomo.stat* фајла који генерише симулатор.

Коришћење импементираног протокола и примењених метрика представљено је примеру анализе перформанси WMN у којој је примењен LQSR протокол рутирања. Сам сценарио симулације подразумева рад са 8 чворова распрострањених у *indoor* окружењу, по угледу на распоред чворова у реалној мрежи која постоји на Електротехничком факултету, Универзитета у Београду. Распоред чворова дат је на слици 3. Параметри сценарија дефинисани су у табели 3.

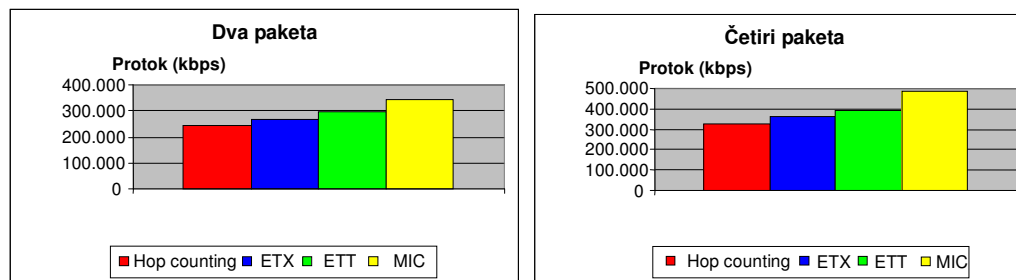


Слика 3. План приземља зграде Електротехничког факултета са распоредом чворова.

Табела 3. Вредности параметара симулације

Трајање симулације	400 s
Протокол рутирања	LQSR/DSR
MAC протокол	MAC/802.11
Пропагациони модел	Indoor
Радна фреквенција	2.4GHz
Капацитет канала	2Mbps
Тип метрике	ETX, ETT, MIC/Hop count
Врста преноса	FTP

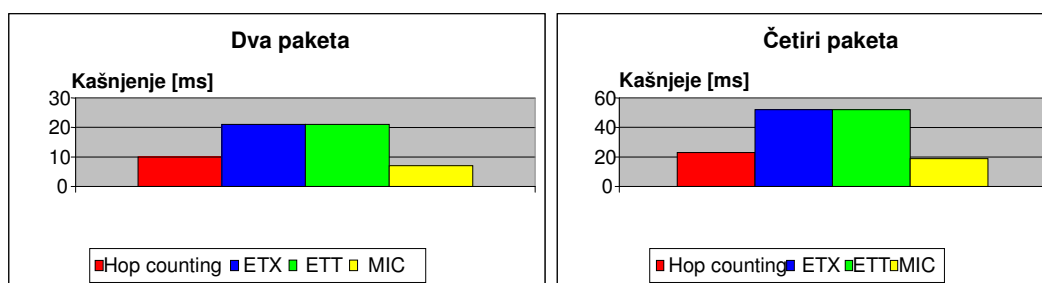
На сликама 4 и 5 приказане су вредности медијане протока и кашњења у мрежи за све наведене метрике, када се шаљу два, и када се шаљу четири пакета.



(a) 2 paketa

(б) 4 paketa

Слика 4. Медијана протока за сваку од четири метрике рутирања када се шаљу два или четири пакета.



(a) 100 чворова

(б) 20 чворова

Слика 5. Просечно кашњење за сваку од наведених метрика када се шаљу два или четири пакета.

Могућности примене

Развијени симулатор је отвореног типа, и погодан је за коришћење и примену у научно-истраживачке и едукативне сврхе. Имплементирани протокол рутирања и метрике засноване су на имплементирању посебно писаног новог програма и његовом повезивању са симулатором. На тај начин ефикасно је комбинован постојећи *open source* симулатор са развојем сопствених програма за симулацију. Избором одговарајућих параметера симулације могуће је извршити различите анализе мрежа са различитим бројем чворова, и различитим саобраћајним оптререћењима. Постојећи симулатор, и протоколи који су у њему имплементирани могу се надоградити сагласно са дефинисаним темама и оствареним резултатима истраживања

Публикације

- [1] Наташа Нешковић, Марија Малнар, “Анализа ETX метрике у реалном пропагационом окружењу mesh мреже“, 27. Симпозијум о новим технологијама у поштанском и телекомуникационом саобраћају – Постел 2009, Београд, децембар 2009.
- [2] Марија Малнар, Наташа Нешковић, “Упоредна анализа Hop counting, ETX, ETT и MIS метрике у реалном пропагационом окружењу једноканалне mesh мреже“, 18. Телекомуникациони форум, ТЕЛФОР 2010, Београд, новембар 2010.

*Софтвер за анализу перформанси WMN у реалом окружењу заснован на мрежном симулатору GloMoSim NS2 је развијен на Електротехничком и Саобраћајном факултету у оквиру текућег пројекта
бр. TP-32025 код Министарства просвете и науке.*

Штампано: јануар 2013.