OTcl софтвер за мрежни симулатор NS2

Руководилац пројекта: Мирјана Стојановић

Одговорно лице: Мирјана Стојановић

Аутори: Мирјана Стојановић, Славица Бошtјанчич Ракас, Валентина Тимченко

Развијено: у оквиру пројекта технолошког развоја ТР-11002

Година: 2008. – 2010.

Примена: 01.10. 2009.

Кратак опис

Развијен је скуп програма (скриптова) за мрежни симулатор NS2 на програмском језику OTcl (*Object-oriented Tool command language*). Дефинисана је општа структура OTcl скрипта, која обухвата: креирање објекта симулатора, управљање помоћним фајловима, формирање топологије мреже, формирање матрице саобраћаја, дефинисање динамике мреже и управљање током симулације. На основу те структуре врши се развој програма за конкретну област и сценарио симулације. Избором одговарајућег OTcl скрипта и подешавањем потребних параметара симулације могуће је, помоћу симулатора NS2, извршити низ различитих анализа у следећим областима: IP квалитет сервиса, *unicast* и *multicast* рутирање саобраћаја, мултипротоколска комутација лабела (MPLS), мобилне ad hoc мреже (MANET). Развијени OTcl скриптови су у потпуности преносиви из оперативног система Windows у Linux и обрнуто.

Техничке карактеристике:
Мрежни симулатор NS2, PC Windows и Linux платформа, објектно оријентисано пројектовање, програмски језик OTcl.

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

Реализатори:
Институт "Михајло Пупин" у Београду

Корисници:
ИМП (интерно), Саобраћајни факултет у Београду, Електротехнички факултет у Београду

Подтип решења:
Софтвер (М85)

Стање у свету

Истраживања у области мрежа са технологијом Интернет протокола (IP) заснивају се на аналитичким методима, симулацији, експериментима и мерењима. Док мерења и експерименти обезбеђују средства за испитивање реалних мрежа, симулација и анализа су ограничене на испитивање конструисаних, апстрактних модела. Ограничење метода мерења и експерименталних метода је у томе што се они могу примењивати само на постојећи систем или делимично нова окружења. Са друге стране, иако су аналитички методи кључни за суштинско разумевање понашања мреже, постоји ризик примене толико поједностављених модела да се губе битне карактеристике понашања мреже.

Симулација је комплементарна са анализом, због тога што омогућава верификацију исправности анализе и испитивање комплексних модела које би било

тешко или немогуће решавати аналитички. Услед хетерогености и брзих промена IP технологије, не постоји јединствен и ограничен скуп сценарија симулације, довољан да покаже да ће се предложени протокол или пројектовани систем добро понашати у условима сталног развоја мреже. Уместо тога, симулације имају битну улогу у испитивању различитих аспеката предложеног решења и разумевању динамичког понашања мреже.

Поред проблема да се дефинише релевантан модел, могу се јавити тешкоће у верификацији да симулатор тачно имплементира жељени модел. Због тога је препоручљиво користити познате и проверене алате, а један од најбољих начина решавања проблема верификације резултата симулације је да симулатори и придружени скриптови буду слободно доступни, тако да други истраживачи могу сами лако да провере ефекте промене полазних претпоставки о мрежном сценарију.

Последњих десетак година је остварен значајан напредак у развоју алата за симулацију мрежа са технологијом Интернет протокола. С обзиром на динамику развоја нових решења заснованих на IP технологији, ови алати се континуирано унапређују и обогаћују новим карактеристикама. Битни захтеви за симулаторе IP мрежа односе се на могућности симулације глобалног Интернета (скалабилност), ефикасно извршавање симулације и уштеду рачунарских ресурса. Већини алата придружени су помоћни графички алати за визуелизацију симулације и обраду и анализу резултата.

Табела 1. Преглед најзаступљенијих симулатора

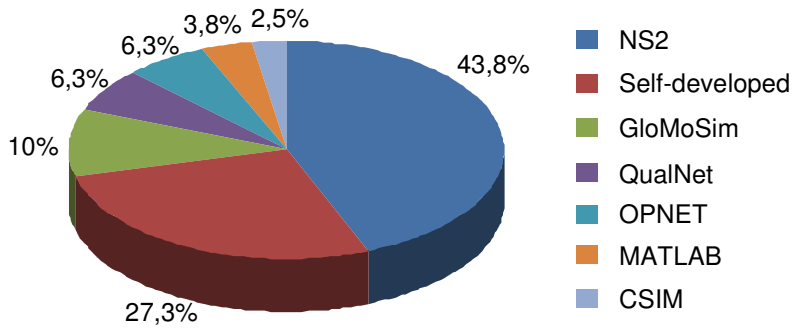
Алат за симулацију и моделовање мрежа	**Доступност**	**URL**
BRITE	*Open-source*	http://www.cs.bu.edu/brite/
Cnet	*Open-source*	www.csse.uwa.edu.au/cnet/
GloMoSim	*Open-source*	http://pcl.cs.ucla.edu/projects/glomosim/
J-Sim	*Open-source*	http://sites.google.com/site/jsimofficial/
NS2	*Open-source*	http://www.isi.edu/nsnam/ns/
OMNeT++	*Open-source*	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	*Open-source*	http://www.ssfnet.org/homePage.html
X-sim	*Open-source*	http://www.cs.arizona.edu/projects/xkernel/

Алати за моделовање и симулацију мрежног слоја обезбеђују корисницима окружење за развој и испитивање мрежних протокола, верификацију дистрибуиране функционалности и анализу перформанси. Основне мере перформанси су кашњење, варијација кашњења (џитер), проценат изгубљених пакета и пропусни опсег

(користан проток). На тржишту доступни симулатори се разликују по цени, једноставности употребе, доступности изворног кода, командног интерфејса, обима доступне документације и других карактеристика. Табела 1 даје преглед симулатора који су најзаступљенији.

Упоредне анализе перформанси постојећих симулатора нису објављене у великом броју, а постојеће публикације углавном обухватају ограничен скуп експеримената са једноставним сценаријима симулације. Осим тога, нису познате публикације које се баве анализом перформанси појединих симулатора на различитим платформама (Windows, Unix).

Студија која је обухватила више од 150 научно-истраживачких радова из области мобилних *ad hoc* мрежа (MANET) статистички је представила заступљеност одређених симултора. Подаци су показали да је већина аутора користила симулатор NS2, као што је приказано на слици 1.



Слика 1. Примењени симулатори у области истраживања MANET мрежа.

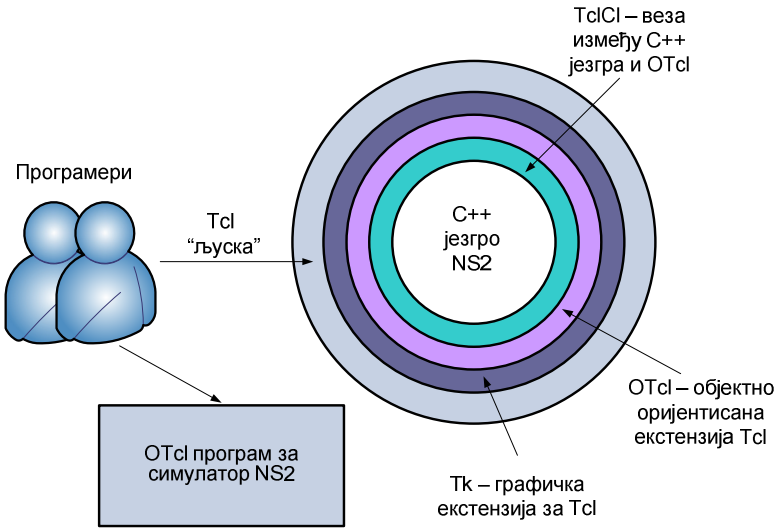
Симулатор NS2 и придружени алати

NS2 припада класи симулатора вођених догађајима у дискретном времену. Развијен је половином деведесетих година у оквиру DARPA VINT (*Virtual InterNetwork Testbed*) пројекта, реализованог у сарадњи неколико водећих америчких универзитета.

NS2 је мулти-протоколски симулатор који имплементира различите апликационе протоколе (Telnet, FTP, HTTP), транспортне протоколе (UDP, више верзија TCP, RTP, RTCP), *unicast* и *multicast* протоколе рутирања, MAC протоколе за симулацију локалних рачунарских мрежа, протоколе за мобилне IP мреже, протоколе за сателитску комуникацију. У NS2 је интегрисана и подршка симулације *Web* апликација са *Web cashing* функцијама и HTTP протоколом. NS2 садржи богат скуп генератора саобраћаја и има могућност рада са узорцима реалног саобраћаја. Симулатор такође обухвата библиотеке за креирање разноврсних топологија мреже и пружа могућности за генерисање компликованијих сценарија симулације, како у погледу скалабилних мрежа, тако и у погледу хетерогених извора саобраћаја. У симулатору су имплементирани механизми квалитета сервиса, као што су алгоритми

за опслуживање пакета, диференцирани сервиси (DiffServ), мултипротоколска комутација лабела (MPLS) и др.

NS2 је развијен методама објектно-оријентисаног пројектовања, на програмском језику C++, са корисничким интерпретером објектно-оријентисаних Tcl (OTcl) скриптова. Расположив је за неколико верзија оперативних система Unix (Sun OS, Free BSD, Linux) и Windows (2000/NT/XP/Vista). Састоји се из пет основних делова: Tcl, Tk, OTcl, TclCl и NS2 *core* (слика 2).



Слика 2. Основни делови NS2 симулатора.

Tcl (*Tool command language*) је истовремено програмски језик и интерпретер релативно једноставне синтаксе, који се ефикасно интегрише са програмским језицима високог нивоа (типично C/C++/C# и Java). Уграђује се у корисничке апликације, а примењује се и за потребе развоја програма за конфигурисање мрежних уређаја (нпр. Cisco Tcl скриптови).

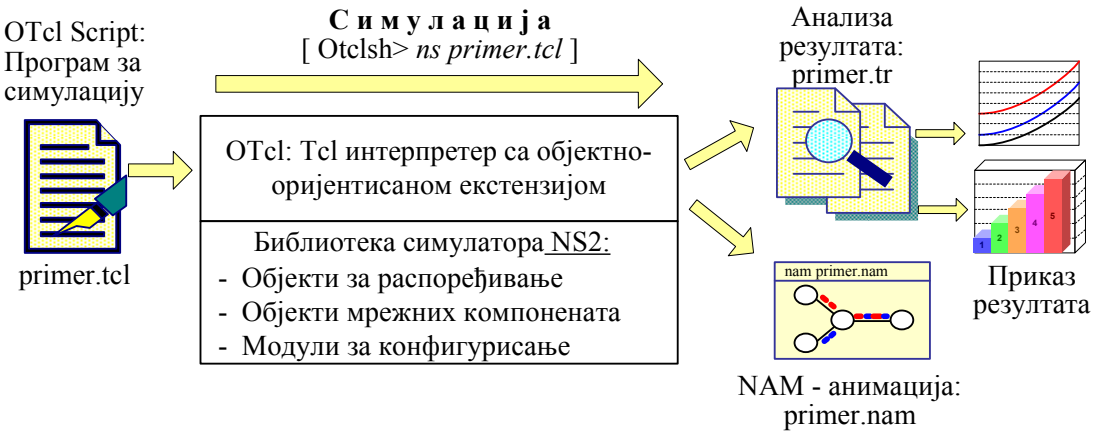
Tk (*Tool kit*) је придружени графички кориснички интерфејс за Tcl.

OTcl (*Object-oriented Tcl*) је објектно-оријентисана екстензија Tcl/Tk, односно кориснички интерпретер објектно-оријентисаних OTcl скриптова. OTcl модули за конфигурисање мреже омогућавају ефикасно успостављање релација између креираних мрежних објеката.

TclCl представља интерфејс између скриптова развијених у OTcl и модула језгра NS2 симулатора развијених на језику C++. Овај интерфејс за сваки C++ објекат креира аналогни OTcl објекат и успоставља потребну контролу.

NS2 *core* представља језгро симулатора генерисано кроз низ C++ модула. Поред различитих протокола и других компоненти мреже, језгро обухвата и модуле за распоређивање догађаја у дискретном времену симулације (*event scheduler*).

На слици 3 је приказан кориснички аспект процеса симулације помоћу NS2. Да би се користио симулатор потребно је програмирати скрипт на језику OTcl. Структура скрипта је детаљније објашњена у поглављу "Скуп OTcl скриптова".



Слика 3. Кориснички аспект процеса симулације.

Дуалност програмских језика је принцип који користе и други симулатори, а потиче од потребе да се повећа ефикасност процеса симулације. У симулатору NS2 се програмирање сценарија обавља помоћу OTcl, који је једноставан за разумевање, учење и примену. Ефикасна обрада генерисаних догађаја омогућена је захваљујући C++ језгру симулатора. Додавање нових компонената симулатору (нпр. нова или модификована верзија неког протокола, нови алгоритам за управљање редовима и др.) врши се допуном изворног C++ кода, а затим компилацијом и линковањем са постојећим библиотекама.

Основне мрежне компоненте симулатора су чвор, линк и пакет. Функције чвора су обрада пакета на основу одговарајућих заглавља (адреса одредишта, адреса извора, тип протокола и др.) и прослеђивање пакета следећем чвору, на основу табеле рутирања и задатог протокола рутирања. У NS2 чвору могу да буду имплементирани различити механизми за управљање редовима (баферима), у зависности од начина имплементације квалитета сервиса. Чворови мреже су међусобно повезани једносмерним или двосмерним линковима, за које се дефинише кашњење услед пропагације и преноса пакета по линку. Линковима се додељују цене – у општем случају, по једна цена за сваки смер комуникације. Такође је могуће симулирати испад линкова у задатим интервалима симулационог времена. NS2 пакет је структура података, која се састоји од скупа заглавља различитих протокола и опционог поља са корисничким подацима.

Симулатор је спрегнут са алатом за анимацију – **NAM** (*Network Animator*), који омогућава визуелно праћење процеса симулације – токова саобраћаја, стања линкова, стања мрежних чворова, стања редова и др.

Анализа резултата симулације обавља се на основу *trace* фајла који генерише симулатор. Генерисање одговарајућих графичких приказа на бази *trace* фајла може

се обављати помоћу неколико слободно доступних алата опште намене, међу којима су познати **Gnuplot** и **Trace Graph** (расположиви за Unix и Windows платформе) и **Xgraph** (расположив за Unix платформе).

Имплементација у окружење реалне мреже доступна је у ограниченом облику, кроз интерфејс који се назива "емулатор" (**NSE**), а расположив је за Unix платформе. За потребе емулације на располагању је екстензија за распоређивање догађаја у реалном времену. Емулатор се може програмирати као генератор и одредиште саобраћаја, када функционише у једном од два режима:

- транспарентном, када нема интеракције са транспортним протоколом, већ су расположиве опције емулације догађаја на нивоу IP пакета и
- протоколском, када емулира TCP транспорни протокол. NSE се интензивно унапређује последњих година, првенствено кроз јавно доступне експерименталне платформе са *Web* корисничким интерфејсима, као што је EMULAB.

Главна предност слободно доступних симулатора за потребе истраживања у области IP мрежа огледа се у могућности додавања нових или измени постојећих карактеристика (предлог нових алгоритама, модификација протокола и др.). Недостатаци NS2 у односу на комерцијалне симулаторе, као што је OPNET Modeler, су лошији кориснички интерфејси и скромније могућности за визуелизацију симулације. Иако је OTcl релативно једноставан програмски језик, учење и овладавање програмирањем скриптова захтева одређено време, а промена карактеристика језгра симулатора захтева добро познавање језика C++ и принципа објектно-оријентисаног пројектовања. Други проблем је документација – у условима сталног развоја симулатора, ажурност документације за NS2 не прати у потпуности реално стање и могућности симулатора.

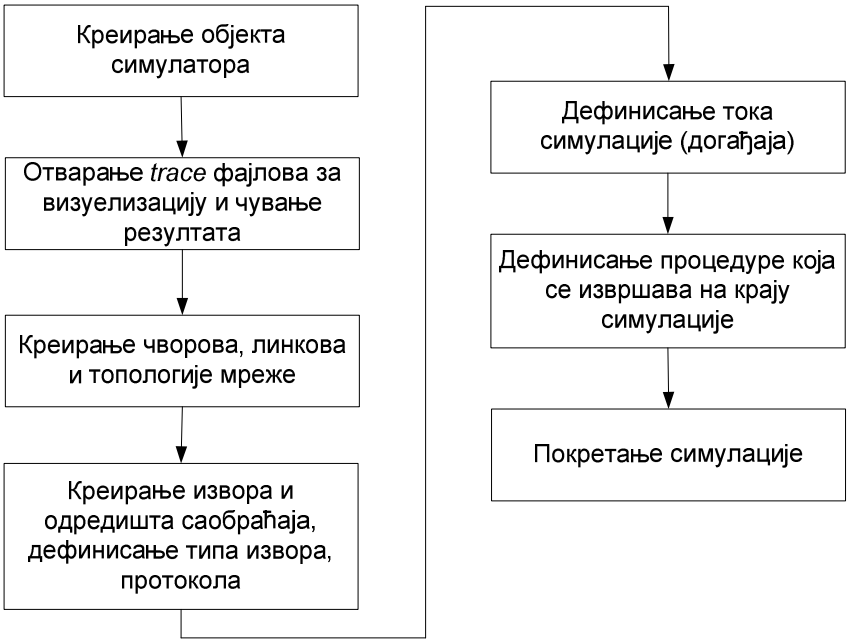
Принцип развоја OTcl скрипта и скуп развијених скриптова

Развој OTcl скрипта

Коришћење NS2 симулатора започиње програмирањем сценарија симулације у виду скрипта на језику OTcl. OTcl скрипт обухвата следеће основне целине: креирање објекта симулатора и иницијализацију распоређивања догађаја, креирање помоћних (*trace*) фајлова за NAM визуелизацију и чување резултата симулације, креирање чворова и топологије/топографије мреже, креирање извора и одредишта саобраћаја, дефинисање протокол стека, дефинисање тока симулације, дефинисање процедура за покретање и завршетак симулације. Основна структура OTcl скрипта приказана је на слици 4.

Сваки OTcl скрипт почиње креирањем објекта симулатора:

```
set ns [new Simulator]
```



Слика 4. Основна структура OTcl скрипта.

Ова наредба генерише објекат симулатора NS2, додељује га променљивој *ns* и чини следеће: иницијализује формат пакета, креира распоређивач догађаја и бира *default* формат адресе.

Објекат симулатора садржи наредбе које:

- креирају сложене објекте као што су чворови и линкови;
- повезују мрежне компоненте креираних објеката (нпр. **`attach-agent`**);
- дефинишу параметре мрежних компонената (већином за сложене објекте);
- повезују агенте одговарајућих протокола (нпр. повезује `tcp` и `tcpsink`).

Следећим наредбама отварају се фајлови (*ns trace* и *nam trace*) у које се даље уписују сви релевантни подаци симулације.

```
set f [open primer.tr w]
      $ns trace-all $f
set nf [open primer.nam w]
      $ns namtrace-all $nf
```

Следећи корак је додавање процедуре `finish` која затвара *trace* фајлове (*ns* и *nam*) и покреће аниматор *nam*.

```
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam primer.nam &
    exit 0
}
```

Процедура `finish` позива се након завршене симулације наредбом

`$ns at time "finish".`

Већина наредби се користи за поставку симулације и распоређивање, међутим неке од њих служе за приказ у NAM-у.

Наредба **`$ns color fid color`** додељује боју току пакета специфицираним са flow id (fid). Ова наредба служи за приказ у NAM-у и нема никакав утицај на стварну симулацију.

Наредбом **`$ns namtrace-all file-descriptor`** симулатор формира запис (*trace*) симулације у NAM формату.

Наредбом **`set n0 [$ns node]`** се креира чвор. Чвор представља сложен објекат NS симулатора, који сачињавају адреса и класификатори одредишта. Корисници могу креирати чвор посебним креирањем адресе и класификатора одредишта и њиховим међусобним повезивањем. Међутим, ова наредба олакшава посао.

Наредба **`$ns duplex-link node 1 node 2 bandwith delay queue-type`** креира два једносмерна линка одређеног пропусног опсега и кашњења и повезује два одређена чвора. У NS2 излазни ред чекања чвора имплементиран је као део линка, зато би приликом креирања линка требало да се дефинише тип реда. У датом примеру тип реда је DropTail.

Као и чвор, и линк је сложен објекат.

Наредба **`$ns queue-limit node 1 node 2 number`** одређује капацитет реда два једносмерна линка који повезују чворове 1 и 2.

Наредба **`$ns duplex-link-op node 1 node 2...`** се користи за приказ у NAM-у.

Након објашњења основне мрежне поставке, следећа ствар је дефинисање саобраћајних агената као што су TCP и UDP, извора саобраћаја као што су FTP и CBR и њихово повезивање са чворовима и агентима, респективно.

Наредба **`set tcp [new Agent/TCP]`** показује како се креира TCP агент. Генерално се сви агенти или извори саобраћаја дефинишу на овакав начин. Да би се креирали агенти или извори саобраћаја морају да се знају имена класа ових објеката (Agent/TCP, Agent/TCPsink, Application/FTP итд.)

Наредба **`$ns attach-agent node agent`** додељује агента чвору, односно позива функцију (*attach-agent*) одређеног чвора која везује дати агент за тај чвор. Нпр. **`$n0 attach $tcp`**.

Након креирања агената који ће комуницирати међусобно, следећа наредба је **`$ns connect agent1 agent2`**, која успоставља логичке мрежне везе између ових агената.

Претпостављајући да је конфигурација мреже готова, следећа ствар је писање сценариа симулације. Постоје многе наредбе које се користе у ову сврху, међутим најчешће се користи следећа:

$*ns* at time "*string*"

Ова наредба дефинише време извршавања одређеног стринга. Нпр. наредба ***$ns* at *0.1* "*$cbr start*"** ће позвати старт функцију CBR извора саобраћаја, која ће покренути CBR да почне са слањем података у тренутку t = 0.1 s.

Након завршене конфигурације мреже, распоређивања и спецификације пост-симулационе процедуре, једина преостала ствар је покретање симулације. То се постиже наредбом ***$ns* run**.

У наставку је приказан карактеристичан пример OTcl скрипта, развијеног за потребе симулације динамичког рутирања у мрежи.

```
#====================================
# OTcl skript:  Primer dinamickog rutiranja
#====================================

# Kreiranje objekta simulatora
set ns [new Simulator]

# Otvaranje ns trace i NAM trace fajlova
set f [open srbija.tr w]
$ns trace-all $f
set nf [open srbija.nam w]
$ns namtrace-all $nf

# Procedura zavrsetka simulacije: zatvaranje trace fajlova i pokretanje nam-a
proc finish {} {
            global ns f nf
                $ns flush-trace
            close $f
                close $nf
                exec nam srbija.nam &
            exit 0
}

# Dinamicko rutiranje (pomocu Distance Vector baziranog protokola
$ns rtproto DV

# Boje tokova saobracaja
$ns color 1 Red
...
$ns color 8 Brown

#====================================
# Definisanje cvorova
#====================================

# Kreiranje  cvorova
for {set i 0} {$i < 31} {incr i} {
        set n($i) [$ns node]
}
```

```
# Definisanje oblika cvorova (default – circle)
$n(15) shape "square"
$n(23) shape "square"
$n(4) shape "square"
$n(7) shape "square"
$n(0) shape "hexagon"

# Definisanje naziva cvorova
$n(0) label BEOGRAD
$n(1) label Pozarevac
$n(2) label Bor
...
$n(30) label Prijepolje

#=====================================
# Formiranje topologije mreze
#=====================================

# Dupleksni linkovi kapaciteta 34Mb/s, kasnjenje 10ms, upravljanje redovima – Drop Tail
$ns duplex-link $n(0) $n(1) 34Mb 10ms DropTail
...
$ns duplex-link $n(26) $n(27) 34Mb 10ms DropTail

# Dupleksni linkovi kapaciteta 155Mb/s, kasnjenje 10ms, upravljanje redovima – Drop Tail
$ns duplex-link $n(0) $n(4) 155Mb 10ms DropTail
$ns duplex-link $n(0) $n(15) 155Mb 10ms DropTail
$ns duplex-link $n(0) $n(7) 155Mb 10ms DropTail

#=====================================
# Definisanje matrice saobracaja
#=====================================

# CBR izvori saobracaja i UDP transportni protokol
# velicina paketa 5000 Byte, interval medjudolazaka – 0.005s
set udp(0) [new Agent/UDP]
$ns attach-agent $n(17) $udp(0)

set cbr(0) [new Application/Traffic/CBR]
$cbr(0) set packetSize_ 5000
$cbr(0) set interval_ 0.005
$cbr(0) attach-agent $udp(0)
$udp(0) set class_ 1

set udp(1) [new Agent/UDP]
$ns attach-agent $n(18) $udp(1)

set cbr(1) [new Application/Traffic/CBR]
$cbr(1) set packetSize_ 5000
$cbr(1) set interval_ 0.005
$cbr(1) attach-agent $udp(1)
$udp(1) set class_ 2

set udp(2) [new Agent/UDP]
$ns attach-agent $n(19) $udp(2)

set cbr(2) [new Application/Traffic/CBR]
$cbr(2) set packetSize_ 5000
$cbr(2) set interval_ 0.005
$cbr(2) attach-agent $udp(2)
$udp(2) set class_ 3
```

```
set udp(3) [new Agent/UDP]
$ns attach-agent $n(16) $udp(3)

set cbr(3) [new Application/Traffic/CBR]
$cbr(3) set packetSize_ 5000
$cbr(3) set interval_ 0.005
$cbr(3) attach-agent $udp(3)
$udp(3) set class_ 4

# FTP izvori saobracaja i TCP transportni protokol
# velicina paketa 1000 Byte, brzina 34Mb/s; default window = 20 paketa
set tcp(0) [new Agent/TCP]
$ns attach-agent $n(26) $tcp(0)
$tcp(0) set class_ 5

set ftp(0) [new Application/FTP]
$ftp(0) attach-agent $tcp(0)
$ftp(0) set type_ FTP

set tcp(1) [new Agent/TCP]
$ns attach-agent $n(28) $tcp(1)
$tcp(1) set class_ 6

set ftp(1) [new Application/FTP]
$ftp(1) attach-agent $tcp(1)
$ftp(1) set type_ FTP

set tcp(2) [new Agent/TCP]
$ns attach-agent $n(4) $tcp(2)
$tcp(2) set class_ 7

set ftp(2) [new Application/FTP]
$ftp(2) attach-agent $tcp(2)
$ftp(2) set type_ FTP

set tcp(3) [new Agent/TCP]
$ns attach-agent $n(25) $tcp(3)
$tcp(3) set class_ 8

set ftp(3) [new Application/FTP]
$ftp(3) attach-agent $tcp(3)
$ftp(3) set type_ FTP

# Definisanje odredista CBR/UDP saobracaja
set null(0) [new Agent/Null]
$ns attach-agent $n(25) $null(0)

set null(1) [new Agent/Null]
$ns attach-agent $n(29) $null(1)

set null(2) [new Agent/Null]
$ns attach-agent $n(26) $null(2)

set null(3) [new Agent/Null]
$ns attach-agent $n(27) $null(3)

# Definisanje odredista FTP/TCP saobracaja
set sink(0) [new Agent/TCPSink]
$ns attach-agent $n(16) $sink(0)

set sink(1) [new Agent/TCPSink]
$ns attach-agent $n(17) $sink(1)
```

```
set sink(2) [new Agent/TCPSink]
$ns attach-agent $n(18) $sink(2)

set sink(3) [new Agent/TCPSink]
$ns attach-agent $n(19) $sink(3)

# Logicko povezivanje odgovarajucih agenata transportnog protokola sa odredistem
$ns connect $udp(0) $null(0)
...
$ns connect $udp(3) $null(3)

$ns connect $tcp(0) $sink(0)
...
$ns connect $tcp(3) $sink(3)

#====================================
# Definisanje toka simulacije
#====================================

# Generisanje saobracaja
$ns at 0.1 "$cbr(0) start"
$ns at 4.5 "$cbr(0) stop"
...
$ns at 0.1 "$ftp(3) start"
$ns at 4.5 "$ftp(3) stop"

# Ispad i oporavak linka
$ns rtmodel-at 1.1 down $n(0) $n(4)
$ns rtmodel-at 4.0 up $n(0) $n(4)

# Promena cena linkova (default =1)
$ns at 1.9 "$ns cost $n(2) $n(3) 100"
$ns at 1.9 "$ns cost $n(3) $n(2) 100"
$ns at 1.9 "$ns cost $n(0) $n(10) 100"
$ns at 1.9 "$ns cost $n(10) $n(0) 100"
$ns at 1.9 "$ns cost $n(7) $n(23) 100"
$ns at 1.9 "$ns cost $n(23) $n(7) 100"
$ns at 1.9 "$ns cost $n(7) $n(0) 100"
$ns at 1.9 "$ns cost $n(0) $n(7) 100"

# Pozivanje procedure za završetak simulacije
$ns at 5.0 "finish"

# Pokretanje simulacije
$ns run
```

Преглед развијених OTcl скриптова

Преглед области истраживања и придружених сценарија симулације приказан је на слици 5. Програмирани су скриптови за анализу контроле загушења, *unicast* и *multicast* рутирања, квалитета сервиса (DiffServ), мобилних и MPLS мрежа. Посебна пажња је посвећена подешавању карактеристика чворова за симулације повезивања фиксних и мобилних чворова где су од велике важности задата међусобна растојања, параметри интерфејса, тип канала и тип пропагације. На основу тога је формиран скуп који садржи више од 100 типских OTcl скриптова, намењених различитим областима истраживања и едукације.

Слика 5. Области истраживања и скуп придружених OTcl скриптова.

Упутство за коришћење софтвера

Симулација

База OTcl скриптова смешта се у директоријум у коме се налази NS2 апликација (извршни фајл симулатора – **`ns.exe`**). Извршавање OTcl скрипта односно покретање NS2 симулатора врши се из програма "Tcl shell" или MS DOS (само за Windows) следећом командом:

```
> ns <naziv_skripta.tcl>
```

Анимација симулације

NAM је графички интерфејс који омогућава приказ дефинисане топологије мреже у изабраном облику који је погодан за визуелно праћење симулације, као и придруживање различитих боја токовима или класама саобраћаја.

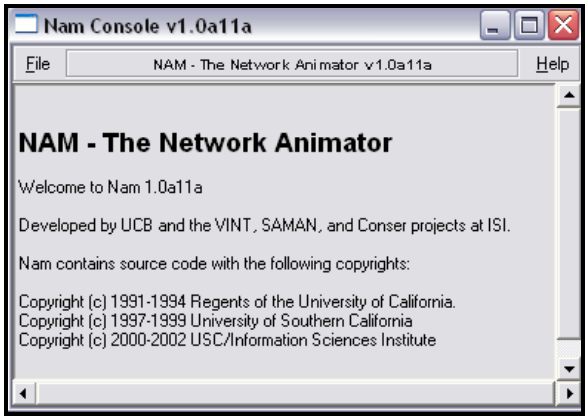
По завршетку симулације, аниматор NAM се покреће аутоматски, под условом да је OTcl скрипт одговарајуће програмиран, као што је приказано у наставку:

```
# Otvaranje NAM trace fajlova
set nf [open primer.nam w]
$ns namtrace-all $nf

# Zatvaranje NAM trace fajla
close $nf

# Komanda za izvršavanje NAM-a
exec nam primer.nam
```

Када је симулација једном извршена, покретање аниматора могуће је следећом командом:

```
> nam <naziv_nam_fajla.nam>
```

Приликом покретања, NAM учита *trace* фајл и генерише два "прозора".



Слика 6. NAM конзола.

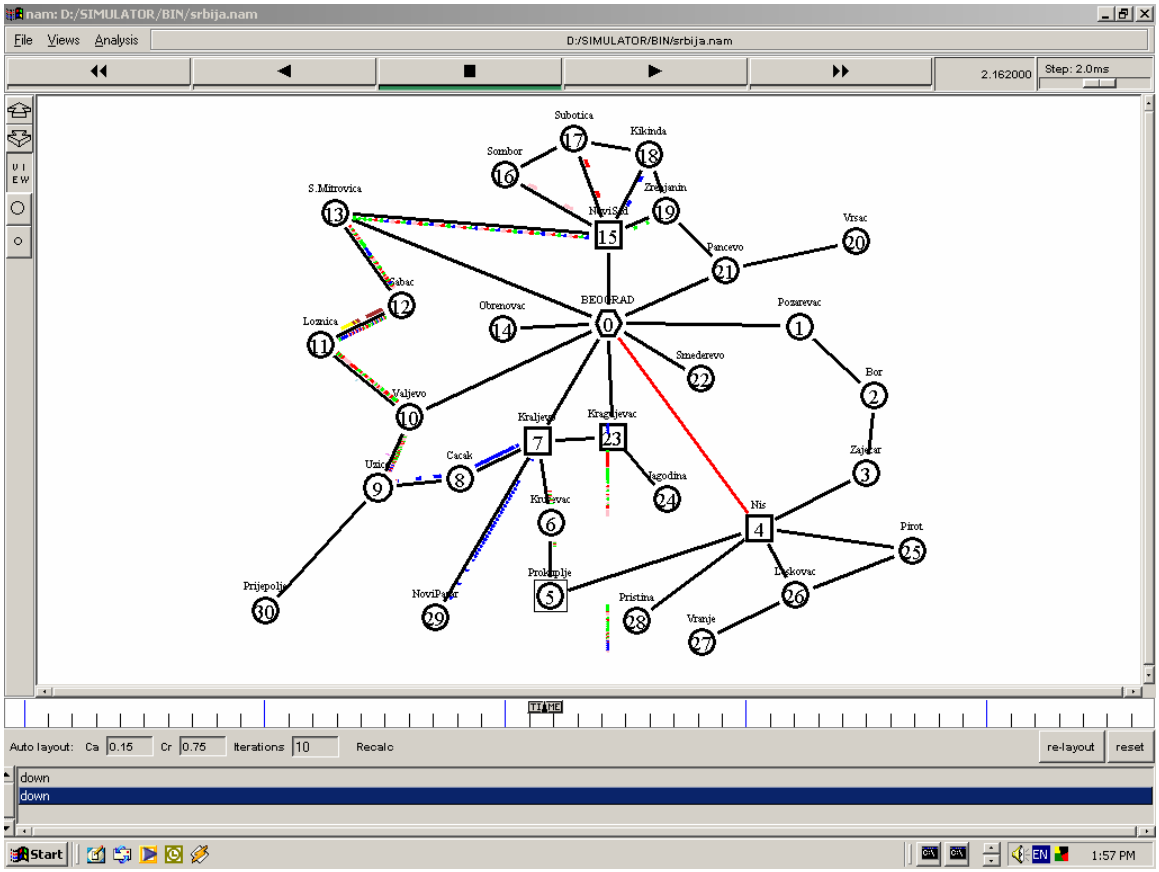
Један од њих је *nam console* прозор, а други представља кориснички интерфејс. Код *nam console* постоје два менија, *File* и *Help* (слика 6).

У оквиру *File* менија постоје следеће опције:

- *Open* омогућава отварање постојећих *nam trace* фајлова, што значи да у једној инстанци аниматора посматрамо више анимација симулације.
- *WinList* приказује листу свих тренутно отворених *trace* фајлова.
- Опцијом *Quit* се програм завршава.

У менију *Help* постоје две опције, у једној је укратко објашњено управљање анимацијом, а у другој су дате информације о ауторским правима и која верзија NAM-а је у питању.

Изглед NAM корисничког интерфејса за симулацију помоћу OTcl скрипта наведеног у претходном поглављу приказан је на слици 7.

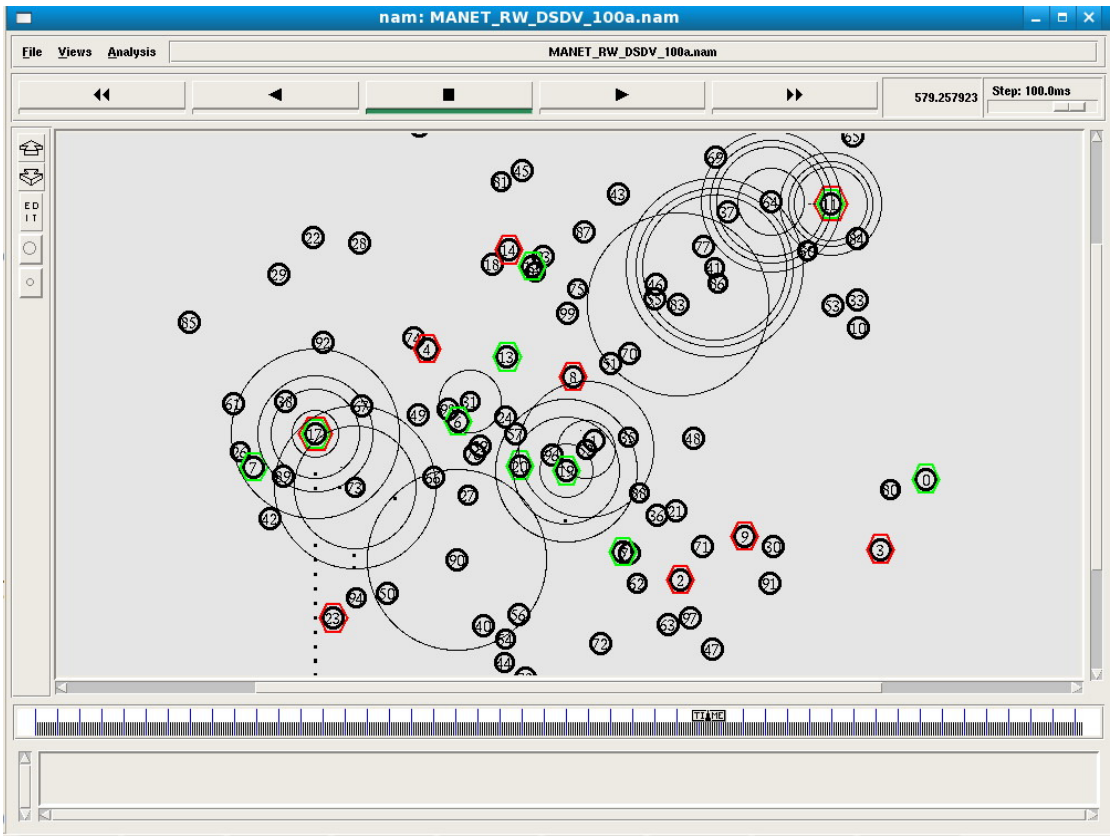


Слика 7. Пример NAM приказа симулиране жичне мреже.

Овај интерфејс је сличан CD *player*-у и омогућава кориснику да покрене, убрза, успори и заустави симулацију или је позиционира на жељено место на временској оси. Такође је имплементирана функција филтрирања – могуће је посматрати одређени тип пакета, мрежни чвор, ток саобраћаја или тип саобраћаја. Функција мониторисања омогућава континуирани надзор вредности специфицираних

променљивих. NAM је користан алат за почетну проверу исправности сценарија симулације програмираног OTcl скриптом: да ли топологија мреже одговара замишљеној, како се усмеравају токови саобраћаја, да ли су коректно дефинисани одговарајући догађаји у времену и др.

Изглед NAM екрана за један пример симулације MANET приказан је на слици 8.

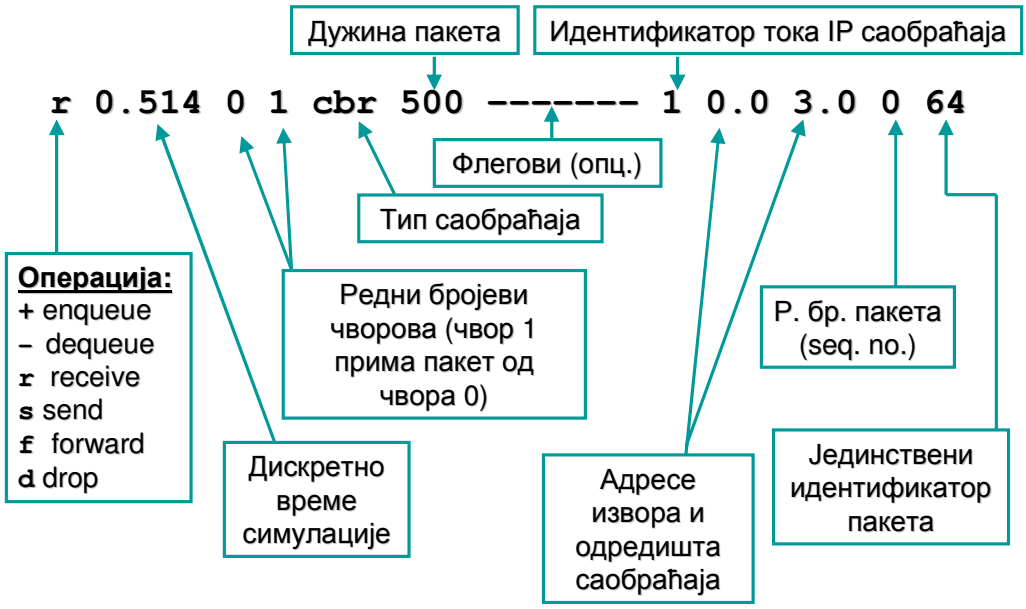


Слика 8. Пример NAM приказа симулиране бежичне мреже (MANET).

Анализа резултата симулације помоћу програма *Trace Graph*

У *trace* фајл, дефинисан у OTcl скрипту, се уписује дефинисани подскуп или целокупан скуп догађаја (генерисање пакета, придруживање реду, прослеђивање, одбацивање пакета и др.) сортиран по дискретном времену симулације.

На слици 9 је приказан формат линије *trace* фајла. Линија почиње дескриптором догађаја симулације (смештање пакета у ред, напуштање реда, пријем, слање, прослеђивање и одбацивање пакета), иза кога следи дискретно време симулације (у секундама). Следећа два дескриптора описују редне бројеве чворова повезане одговарајућим линком. Следеће информације су тип и величина пакета. Следи поље за флегове који су опциони. Поље иза флегова представља идентификатор тока IP саобраћаја. Наредна два поља представљају адресе извора и одредишта. Последња два поља представљају редни број пакета и јединствени идентификатор пакета.

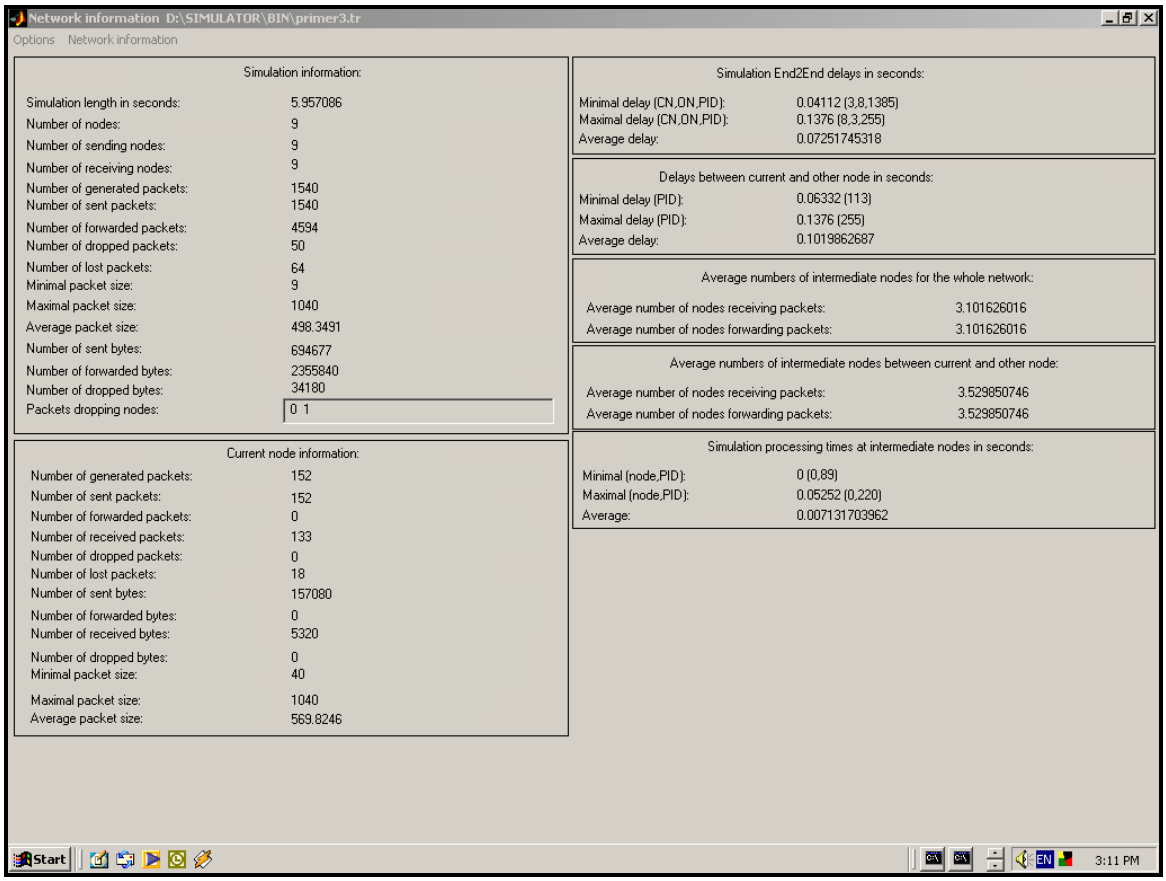


Слика 9. Формат линије *trace* фајла.

Trace Graph је софтвер, расположив за Linux и Windows платформе, специфично развијен за обраду и анализу NS2 *trace* фајлова. Мени са информацијама о мрежи (*Network Information*) омогућава приказ и меморисање следећих информација:

1. **Ток симулације**, што обухвата трајања симулације, број прослеђених пакета, број прослеђених битова, број изгубљених пакета итд.
2. **Посматрани чвор**
   Посматрани чвор (*current node*) је извор а други чвор (*other node*) је одредиште када се ради о генерисаним/послатим/прослеђеним пакетима одн. битовима. Посматрани чвор је одредиште а други чвор извор када се ради о примљеним/изгубљеним пакетима одн. битовима.
   Број изгубљених пакета представља разлику броја прослеђених пакета посматраном чвору и броја примљених пакета другог чвора ако је изабрана опција *direct connection*. У супротном случају то је разлика послатих пакета посматраног чвора и број примљених пакета другог чвора.
3. **Кашњење између крајњих тачака** (у једном смеру и у оба смера – директном и повратном)
   Кашњење између крајњих тачака (*End2End delay*) представља разлику времена (у секундама) када је пакет примио други чвор и времена када је посматрани чвор пакет послао.
4. **Кашњење између два специфицирана мрежна чвора**
   Ова информација се приказује само у случају да је избрана опција *other node*. Приказује *End2End* кашњење (у секундама) између посматраног (извор) и другог (одредиште) чвора када није изабрана опција *direct connection*. У супротном приказује кашњење између посматраног (прослеђује пакете) и другог (прима пакете) чвора.
5. **Просечан број транзитних чворова** (између извора и одредишта) у мрежи
6. **Просечан број транзитних чворова** између два специфицирана чвора
7. **Време обраде пакета у чвору**.

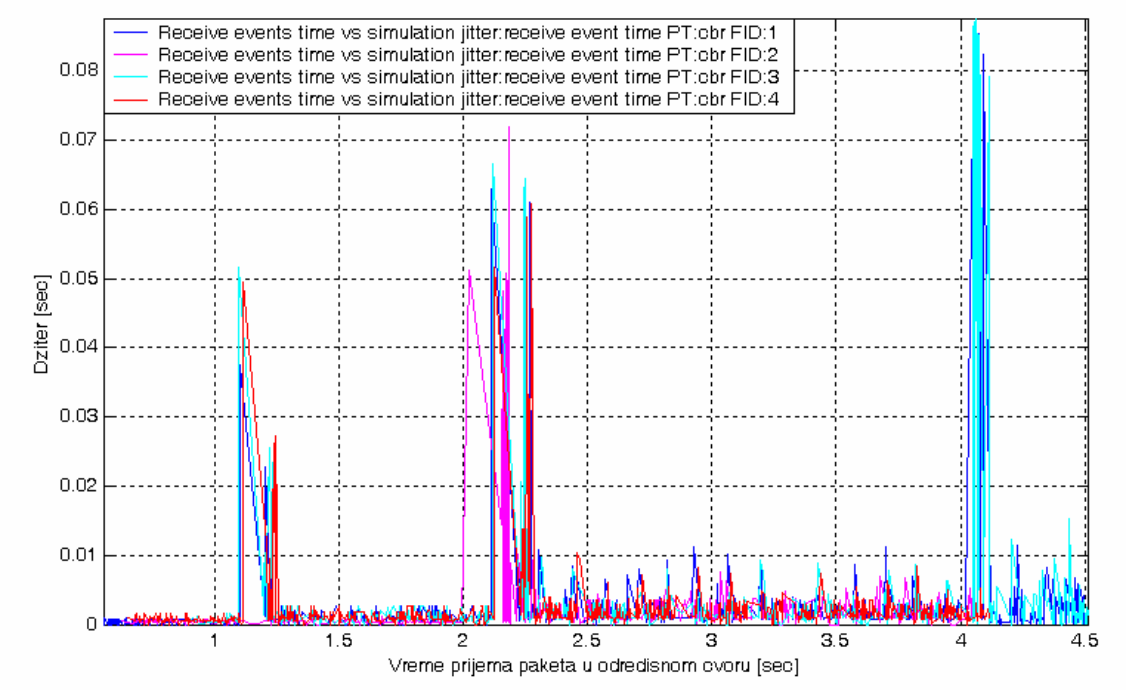
Пример прозора са информацијама о мрежи симулираној помоћу OTcl скрипта наведеног у претходном поглављу приказан је на слици 10.



Слика 10. Информације о симулацији, чворовима и кашњењу.

Tracegraph омогућава и да корисник дефинише интервал времена у коме жели да посматра симулационе догађаје, да меморише информације у ASCII (txt) формату, да меморише графике (JPEG и TIFF формати), да бира између различитих опција приказа графика, да анализира догађаје између дефинисаног пара чворова (извор-одредиште), да анализира специфицирани ток пакета и др.

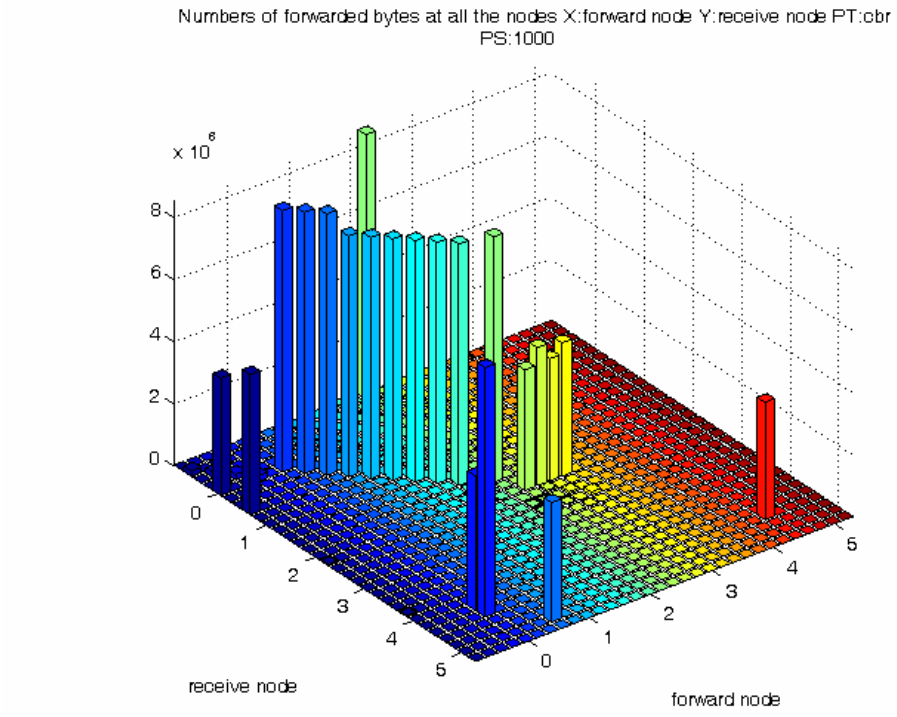
Дводимензионални (2D) графички приказ резулата обухвата широк скуп разноврсних функција: пропусна моћ у функцији кашњења, џитера или времена обраде пакета, џитер у функцији времена генерисања, прослеђивања, пријема или одбацивања пакета, разне кумулативне функције расподеле, хистограми и др. 2D графици могу се зумирати и то левим или десним тастером миша. Постоји и опција преклапања графика (*Overlay graph*) која омогућава истовремени приказ до седам 2D графика. На слици 11 је приказан пример дводимензионалног приказа варијације кашњења (џитер) у односу на време пријема пакета четири тока саобраћаја (симулација помоћу OTcl скрипта наведеног у претходном поглављу).



Слика 11. Trace Graph: пример 2D приказа резултата симулације

.

Тродимензионални (3D) графички приказ резултата обухвата функције броја генерисаних, примљених, прослеђених или одбачених пакета (бајтова) у функцији комбинација изворних, одредишних и транзитних чворова. 3D графици могу се и ротирати ради прегледнијег приказа. Пример 3D приказа броја прослеђених бајтова у сваком чвору мреже приказан је на слици 12.



Слика 12. Trace Graph: пример 3D приказа резултата симулације.

Trace Graph користи библиотеке програмског пакета MATLAB. NS2 *trace* фајл се може меморисати у форми "*.mat" (MATLAB) фајла ради ефикасније поновне обраде и анализе резултата.

Анализа резултата симулације помоћу програма *Xgraph*

NS2 Linux верзија за потребе графичке интерпретације добијених *trace* фајлова користи напредни *Xgraph* програм који долази у пакету са NS2 инсталацијом. *Xgraph* нема своју Windows верзију. Рад *Xgraph* апликације се заснива на очитавању жељених података из *trace* фајла у складу са претходно генерисаним скриптом `.awk` типа. `Awk` је екстензија за скриптове којима дефинишемо шта нам је од информација из целог *trace* фајла потребно и те информације на уређен начин издвајамо. На основу сазнања колико траје симулација, на којим местима у *trace* фајлу се налазе жељене информације и избора типа пакета који корисник жели да анализира, могуће је генерисати посебан део скрипта који ће *trace* фајл обрадити на задати начин. У наставку је приказан пример `.awk` скрипта који је примењен за потребе анализе резултата симулације трајања 150 секунди и примењених CBR извора саобраћаја.

```
BEGIN{
   sim_end = 150;
   i=0;
   while (i<=sim_end) {sec[i]=0; i+=1;};
}
{
   if ($1=="r" && $7=="cbr" && $3=="_1_") {
          sec[int($2)]+=$8;
    };
}
END{
   i=0;
   while (i<=sim_end) {print i " " sec[i]*8; i+=1;};
}
```

На основу овог скрипта је омогућено прорачунавање протока података за задати чвор **_i_** током целе симулације. Потребно је израчунати збир свих величина пакета које чвор **_i_** прима. Резултат је низ вредности оствареног протока за сваку секунду симулације.

Перформансе симулатора NS2 у окружењима Windows и Linux

Компаративна анализа карактеристика симулатора IP мрежа NS2 при раду у Linux и Windows окружењима подразумева експериментално поређење које се заснива на програмирању већег броја OTcl скриптова и њиховом извршавању на оба оперативна система. Конфигурација хардвера и софтвера коришћена при тестирању приказана је у табели 2.

Табела 2. Спецификације хардверског окружења, оперативних система и симулатора

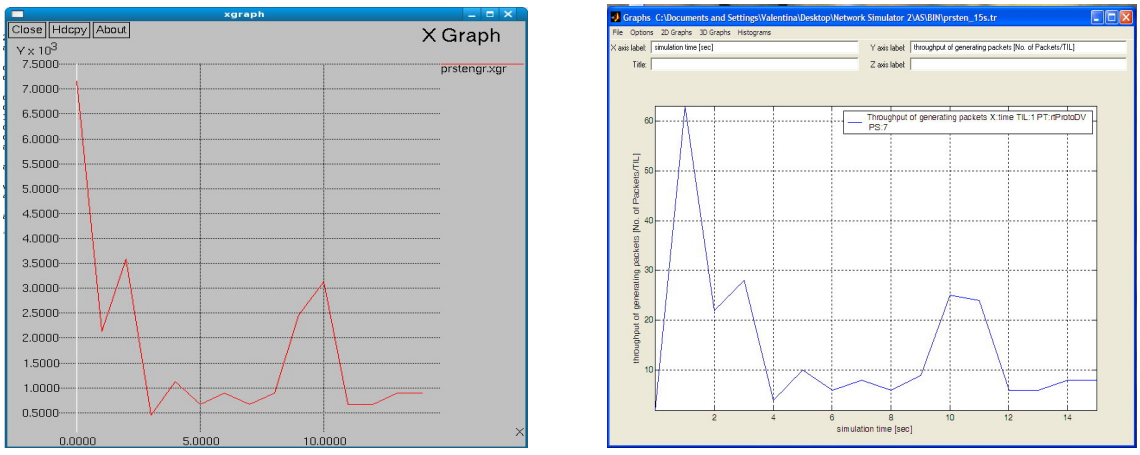
Хардверска конфигурација	**Оперативни системи**	**Симулатор**
Pentium 4 CPU 1,8 GHz RAM 480 MB	Windows XP Professional (верзија 2002, SP2)	NS2v2.32 NAMv1.13 Tclv8.4.15
	Linux, RedHat дистрибуција Fedora 8, kernel 2.6	

Инсталација NS2 софтвера у Linux окружењу се може реализовати на два начина:

- компилацијом изворних програма из пет основних делова (OTcl, графички интерпретер OTcl скриптова, објектно оријентисана екстензија – OTcl, интерфејс OTcl и C++, језгро симулатора) и затим линковањем у извршну верзију;
- инсталацијом *all-in-one* пакета, што подразумева отпакивање пакета у одговарајућем фолдеру а затим у Терминалу покретање инсталације командом `./install`.

Иако је у Windows окружењу предвиђена могућност инсталације *all-in-one* пакета помоћу програма *Cygwin*, искуство је показало да се коректна инсталација може добити ако се сваки део програма инсталира посебно, а затим линкује у извршну верзију. Овај метод је компликованији и захтева добро познавање C++ развојног окружења. Након инсталације се обавља верификација кроз пробно тестирање над специјално генерисаним скриптом.

Рад аниматора NAM и графички кориснички интерфејси су исти у Windows и Linux окружењу. Компаративна анализа рада NS2 у ова два оперативна система има своју димензију у контексту алата за обраду резултата.



Слика 13. Xgraph и Trace Graph: пример приказа резултата симулације.

NS2 Linux верзија има могућност коришћења апликације *Xgraph*, за потребе графичке интерпретације резултата. Примена *Xgraph* апликације захтева од

корисника детаљно познавање структуре *trace* fajla. На слици 13 су приказани еквивалентни графици, генерисани помоћу програма *Xgraph* и *Trace Graph.*

Резиме упоредне анализе перформанси симулатора NS2 у окружењима Windows и Linux приказан је у табели 3.

Развијени скуп OTcl скриптова је у потпуности преносив из једног у други оперативни систем. Једини изузетак представљају делови скрипта који се односе на покретање програма *Xgraph*, који су применљиви само под оперативним системом Linux (они се једноставно могу деактивирати ако се захтева примена у Windows окружењу).

Табела 3. Резиме карактеристика симулатора NS2 у окружењима Windows и Linux

Карактеристика	**Windows**	**Linux**
Инсталација	Из делова	*All-in-one* или из делова
Кориснички интерфејс	Већина корисника на почетку истраживачког рада користи NS2 у Windows окружењу. Прегледно окружење за потребе едукације и приказа нових решења.	У случају сложенијих симулација корисницима се предлаже рад у Linux окружењу
Програмерско искуство	Основно коришћење NS2 захтева овладавање програмирањем у OTcl програмском језику, доста једноставнијем од C++. У случају потребе за променом карактеристика језгра симулатора неопходно је добро програмерско искуство (принципи објектно оријентисаног пројектовања, C++).	
Анимација	NAM (остварује сличне перформансе у оба окружења)	
Анализа резултата	*Trace Graph*	*Xgraph*, *Trace Graph*
Резултати симулације фиксних мрежа	Спорији; заузима више ресурса	Брже симулације; заузима мање ресурса
Резултати симулације MANET	Проблем склабилности: до 100 чворова без фиксне инфраструктуре; до 30 чворова са фиксном инфраструктуром; Проблем анализе резултата помоћу програма *Trace Graph*	Извршава симулације мрежа са више од 100 чворова; *Trace Graph* се без проблема примењује у анализи добијених резултата

Интеграција са софтвером BonnMotion

Генерисање модела мобилности у окружењу симулатора подразумева да се у потпуности дефинишу сви параметри карактеристични за одговарајуће симулирано окружење и примену симулиране мреже.

У истраживањима MANET мрежа примењен је приступ заснован на коришћењу софтвера *BonnMotion* због великог броја модела мобилности и придружених параметара за подешавање.

BonnMotion је програм развијен на програмском језику Java, у оквиру групе *Communication Systems* Универзитета у Бону. Користи се за генерисање модела мобилности у окружењу симулатора. Компатибилан је са NS2 и GloMoSim/QualNet мрежним симулаторима тако да се генерисани сценарији мобилности могу интегрисати у симулације ова два симулатора. *BonnMotion* подржава генерисање следећих модела мобилности: RW (*Random Waypoint*), GM (*Gauss Markov*), MG (*Manhattan Grid*), RPGM (*Reference Point Group Mobility*) и статички.

Софтвер је слободно доступан и омогућава креирање адекватног симулационог окружења у складу са корисничким спецификацијама. С обзиром да је заснован на програмском језику Java, да би могао да се користи, неопходна је инсталација JDK (*Java Development Kit*) или JRE (*Java Runtime Environment*). Подржане су инсталације на платформама UNIX и Windows (кроз *Cygwin*).

У оквиру *BonnMotion* пакета се налази неколико апликација. Осим апликације за генерисање модела мобилности, укључене су апликације које омогућавају статистичку обраду података и анализу карактеристика сценарија. Софтвер има интегрисан **`help`** фајл који се покреће из терминала. За покретање сваке од апликација користи се специфична команда:

```
./bm <parameters> <application> <application parameters>
```

Постоје два начина уноса података везаних за модел мобилности. Параметри се могу унети кроз командну линију или помоћу генерисаног фајла у којем ће бити дефинисани параметри и њихове вредности. Ове две методе се у случају потребе могу комбиновати јер параметри унешени преко командне линије имају предност у односу на оне унете преко генерисаног фајла и могу их поништити.

Скриптови *BonnMotion* софтвера се покрећу у терминалу. На основу команде се задају карактеристике жељеног модела мобилности. Пример основне команде је дат у наставку:

```
./bm -f <sc_name> <model> -n <num> -d <dur> -i <incph> -x <x> -y <y>
```

Овом командом ће генератор креирати нови сценарио под именом **`sc_name`**, дефинисан моделом мобилности **`model,`** и бројем чворова дефинисаним у променљивој **`num`**. Трајање симулације се дефинише вредношћу променљиве **`dur`** изражено у секундама, док се иницијална фаза која се одбацује дефинише кроз променљиву **`incph`** и такође је изражена у секундама. Област симулације се дефинише кроз параметре **`x`** и **`y`** који су изражени у метрима.

Да би генерисани фајл за неки сценарио мобилности могли да применимо у NS2 мрежном симулатору, неопходно је обавити конверзије у одговарајући формат. Команда:

```
./bm NSFile –f <sc_name>
```

ће генерисати фајл **sc_name.ns_movements** који је скрипт верзија погодна за примену у NS2 окружењу. Овај скрипт се уводи у главни OTcl симулациони скрипт.

Примена

Примена у истраживању

Развијени софтвер је примењен у Институту "Михајло Пупин" за потребе истраживања у области IP квалитета сервиса (*Quality of Service*, QoS) и анализи перформанси мобилних ad hoc мрежа (MANET).

Истраживања IP квалитета сервиса су усмерена на QoS архитектуре и механизме, као и оптимизацију перформанси оперативне мреже (инжењеринг саобраћаја). Симулација се примењује, самостално и у комбинацији са аналитичким методима, у свим елементима избора решења за имплементацију QoS, што обухвата моделовање архитектуре QoS, варијанте класификације саобраћаја, избор и конфигурисање механизама опслуживања пакета, управљање квалитетом сервиса, QoS рутирање.

Основни циљеви истраживања у области MANET обухватили су испитивање принципа конструисања симулационих модела, као и развој, имплементацију и тестирање скупа модела у окружењу мрежног симулатора NS2. Најважнији резултати истраживања су:

- Предлог приступа генерисању сценарија мобилности, заснован на коришћењу софтвера BonnMotion. На тај начин су ефикасно комбиновани постојећи *open source* алати са развојем сопствених програма за симулацију.
- Развој модела симулације, на основу кога су извршена два примера анализе перформанси, која су указала на оптималне начине примене метода симулације у истраживању MANET. Анализа је обухватила више од 200 изведених симулација помоћу карактеристичних OTcl скриптова, развијених за потребе коришћења NS2/BonnMotion платформе.

Примена у едукацији

Визуелним приказом динамике мреже помоћу аниматора NAM, могуће је боље објаснити проблематику из области телекомуникационих и рачунарских мрежа. За ту сврху користе се карактеристични OTcl скриптови из описаног скупа. Области у којима је NAM ефикасно примењиван као допуна слајдовима/предавањима на табли су: илустрација функција појединих слојева OSI модела, *unicast* рутирање (алгоритми Dijkstra и Bellman-Ford), multicast рутирање, механизми и верзије TCP протокола, архитектура диференцираних сервиса, рутирање саобраћаја у MPLS мрежи и др. Овај приступ се примењује, почевши од 2008. године, у настави из

следећих предмета: *Телекомуникационе мреже*, *Широкопојасне телекомуникационе мреже* и *Протоколи у телекомуникационим мрежама* (Електротехнички факултет у Београду) и *Рачунарске мреже* (Саобраћајни факултет у Београду) .

Студенти ових факултета такође примењују описани скуп OTcl скриптова за потребе израде семинарских, дипломских и мастер радова, као и магистарских теза. Софтвер је отвореног типа, а постојећи скуп скриптова се перманентно надограђује, сагласно дефинисаним темама и остварeним резултатима истраживања.

Публикације

[1] В. Тимченко, Принципи симулације мобилних ad hoc мрежа, магистарска теза, Електротехнички факултет Универзитета у Београду, март 2010.

[2] В. Тимченко, Б. Ђорђевић, "Модели мобилности у симулацији MANET мрежа", Зборник радова *Телекомуникационог форума ТЕЛФОР 2009 (CD)*, Београд, новембар 2009, стр. 1331-1334.

[3] V. Timčenko, M. Stojanović, S. Boštjančič Rakas, "MANET Routing Protocols vs. Mobility Models: Performance Analysis and Comparison", in *Recent Advances in Applied Informatics and Communications, Proceedings of the 9th WSEAS International Conference on Applied Informatics and Communications (AIC'09)*, N. E. Mastorakis, M. Demiralp, V. Mladenov, Z. Bojković (eds), WSEAS Press, Moscow, August 2009, pp. 271-276.

[4] V. Aćimović-Raspopović, M. Stojanović, J. Teodorović, "The Undergraduate Training on Simulating IP Networks Using Network Simulator NS2", *Proceedings of the XV International Symposium on Theoretical Engineering – ISTET 2009 (CD)*, Lübeck, Germany, June 2009, pp. 216-219.

[5] М. Стојановић, С. Боштјанчич Ракас, В. Тимченко, "Искуства у едукацији и истраживању IP мрежа методом симулације", рад по позиву, *Зборник радова XXVI Симпозијума о новим технологијама у поштанском и телекомуникационом саобраћају – ПОСТЕЛ 2008*, Београд, децембар 2008, стр. 267-278.

[6] V. Timčenko, S. Boštjančič Rakas, M. Stojanović, "Analiza karakteristika simulatora IP mreža NS2 u Linux i Windows okruženju", Zbornik radova 16. Telekomunikacionog foruma – TELFOR 2008 (CD), Beograd, novembar 2008, str. 146-149.

OTcl софтвер за мрежни симулатор NS2 је развијен у Институту Михајло Пупин у оквиру текућег пројекта бр. ТР-11002 код Министарства за науку и технолошки развој.

Штампано: април 2010.