

Analiza karakteristika simulatora IP mreža NS2 u Linux i Windows okruženju

Valentina Timčenko, Slavica Boštjančič Rakas, Mirjana Stojanović

Sadržaj — U radu je prikazana komparativna analiza karakteristika simulatora IP mreža NS2 pri radu u Linux i Windows okruženjima. Eksperimentalno poređenje zasniava se na programiranju većeg broja OTcl skriptova simulacije i testiranju rada simulatora na oba operativna sistema. Prikazani su početni rezultati analize, kojom su obuhvaćeni sledeći aspekti: instalacija, korisnički aspekt, animacija, alati za analizu rezultata, poređenje scenarija simulacije i skalabilnost.

Ključne reči — Internet protokol, Linux, NS2, simulacija, Windows

I. UVOD

SVAKI pokušaj simulacije ponašanja globalnog Interneta predstavlja pravi izazov, a ključni problem koji otežava modelovanje i simulaciju predstavlja upravo velika heterogenost globalne mreže. Ona se ogleda u različitosti topologija mreža, generisanja saobraćaja, razlika u protokolima i njihovim verzijama kao i raznovrsnosti korisničkih aplikacija.

Simulacije Interneta su najkorisnije za razumevanje dinamike mreže, ilustraciju poente istraživanja i istraživanje neočekivanih ponašanja (npr. “worst case”) [1]. Simulacije mogu biti nesigurne kada se koriste samo da rezultuju kvantitativnim vrednostima; umesto toga bitno je ustanoviti: (1) da li male promene modela prouzrokuju značajne promene rezultata; (2) da li bi rezultati bili promenjeni usled promene detalja softverske implementacije simulatora i (3) da li su rezultati skalabilni.

Tokom poslednjih desetak godina ostvaren je veliki napredak u razvoju alata za simulaciju Interneta i IP baziranih mreža. S obzirom na dinamiku razvoja novih rešenja zasnovanih na IP tehnologiji, ovi alati se kontinuirano unapređuju i obogaćuju novim karakteristikama. Bitni zahtevi za simulator IP mreža su: mogućnosti simulacije skalabilnih mreža, efikasno izvršavanje simulacije i ušteda računarskih resursa.

Primeri poznatih simulatora IP mreža su: NS2 (Network Simulator ver. 2), OPNET Modeler, SSFNet i J-Sim [2].

V. Timčenko, Institut Mihajlo Pupin, Volgina 15, 11060 Beograd, Srbija (e-mail: valentina@kondor.imp.bg.ac.yu).

S. Boštjančič Rakas, Institut Mihajlo Pupin, Volgina 15, 11060 Beograd, Srbija; (e-mail: slavica@kondor.imp.bg.ac.yu).

M. Stojanović, Institut Mihajlo Pupin, Volgina 15, 11060 Beograd, Srbija; (e-mail: stojmir@kondor.imp.bg.ac.yu).

U poređne analize performansi navedenih simulatora nisu objavljene u velikom broju, a postojeće publikacije uglavnom obuhvataju ograničen skup eksperimenata sa jednostavnim scenarijima simulacije [3], [4]. Većina ovih alata dostupna je za različite softverske platforme (Windows, Unix). Autorima ovog rada nisu poznate publikacije koje se bave analizom performansi pojedinih simulatora na različitim platformama.

Motivacija autora da se posvete komparativnoj analizi performansi simulatora NS2 u Linux i Windows okruženju podstaknuta je istraživanjima u oblasti mobilnih ad hoc mreža (*Mobile Ad hoc NETWORK* – MANET), kada smo se suočili sa problemima sa NS2 instalacijom na Windows platformi. U ovom radu su predstavljeni početni rezultati analize, kojom su obuhvaćeni sledeći aspekti: instalacija, korisnički aspekt, animacija, alati za analizu rezultata, dobijeni rezultati simulacije i skalabilnost.

II. MREŽNI SIMULATOR NS2

Simulator NS2 [5] je razvijen polovinom devedesetih godina u okviru DARPA VINT (*Virtual InterNetwork Testbed*) projekta, realizovanog u saradnji nekoliko vodećih američkih univerziteta. Implementira niz različitih aplikacionih i transportnih protokola, unicast i multicast protokole rutiranja, protokole pristupa medijumu (MAC), protokole za mobilne mreže kao i za satelitsku komunikaciju. Najčešću primenu je našao u istraživanjima LAN i IP WAN mreža, ad-hoc, mobilnih i satelitskih mreža. Takođe obuhvata mehanizme kvaliteta servisa (opsluživanje paketa i upravljanje redovima, diferencirane servise), multiprotokolsku komutaciju labela (MPLS) i dr.

NS2 je simulator vođen događajima u diskretnom vremenu. Razvijen je metodama objektno-orijentisanog projektovanja, na programskom jeziku C++, sa korisničkim interpreterom objektno-orijentisanih OTcl skriptova (*Object-oriented Tool Command Language* – OTcl). Korišćenje dva programska jezika potiče od potrebe da se poveća efikasnost procesa simulacije. Programiranje scenarija simulacije obavlja se pomoću OTcl, koji je relativno jednostavan za razumevanje, učenje i primenu. Efikasna obrada generisanih događaja omogućena je zahvaljujući C++ jezgri simulatora.

NS je *open source* softver, koji je prvobitno bio razvijen za UNIX platforme (FreeBSD, Linux, SunOS, Solaris). Međutim veliki broj korisnika Windows (98/2000/NT/XP) platforme zahtevao je prilagodavanje aplikacije za ovo okruženje tako da ga danas možemo naći

u obe varijante. Najnovija verzija, NS2 je 2.33. Vizuelni prikaz scenarija i toka simulacija omogućen je kroz NAM (*Network Animator*) aplikaciju, a analiza dobijenih rezultata može da se obavi pomoću različitih aplikacija koje omogućavaju grafički prikaz ispitivanih performansi mreže.

III. ANALIZA PERFORMANSI

Radno okruženje predstavljaju: (1) Windows XP Professional, version 2002, SP2 i (2) Linux, koji je specificiran kernelom 2.6 na RedHat distribuciji i Fedora Linux verzijom 6. Svi testovi izvršeni su na Pentium 4 CPU, 1.8GHz i 480MB RAM. Ispitivanje karakteristika rada NS2 u ova dva okruženja izvedeno je na osnovu izvršavanja niza simulacija istih skriptova na jednom, a zatim i na drugom operativnom sistemu. Posebna pažnja je poklonjena fiksnim mrežama, DiffServ simulacijama i MANET simulacionim modelima. Nakon izvršene NS2 simulacije generiše se *trace* fajl. Obrada tako dobijenog *trace* fajla vrši se pomoću programa Trace Graph [6] (Windows i Linux) i Xgraph (Linux).

A. Instalacija

Instalacija NS2 softvera u Linux okruženju se može realizovati na dva načina: (1) kompilacijom izvornih programa iz pet osnovnih delova (OTcl, grafički interpreter OTcl skriptova, objektno orijentisana ekstenzija OTcl – OTcl, interfejs OTcl i C++, jezgro simulatora) i zatim linkovanjem u izvršnu verziju i (2) u takozvanom “all-in-one” paketu, što podrazumeva otpakivanje paketa u odgovarajućem folderu a zatim u Terminalu pokretanje instalacije komandom `./install`. Iako je u Windows okruženju predviđena mogućnost instalacije “all-in-one” paketa pomoću programa Cygwin, iskustvo autora ovog rada je da se korektna instalacija može dobiti ako se svaki deo programa instalira posebno, a zatim linkuje u izvršnu verziju. Ovaj metod je komplikovaniji i zahteva dobro poznavanje C++ razvojnog okruženja.

Nakon instalacije se obavlja verifikacija kroz probno testiranje nad specijalno generisanim skriptom.

B. Korisnički aspekt

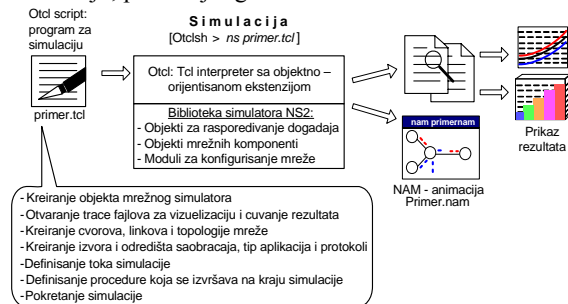
NS2 simulator mreža se danas koristi u velikom broju akademskih institucija u svetu. Njegova prvenstvena primena je za potrebe istraživanja i razvoja novih rešenja u oblasti IP baziranih mreža. Simulator se takođe može koristiti za potrebe edukacije i prikaza novih rešenja.

Najčešći problem koji se javlja kod korisnika je teškoća u učenju usled nepotpune dokumentacije. Iako je OTcl relativno jednostavan programski jezik, učenje i ovladavanje programiranjem skriptova zahteva određeno vreme, a promena karakteristika jezgra simulatora zahteva dobro poznavanje jezika C++ i principa objektno-orijentisanog projektovanja.

Slika 1 daje prikaz korisničkog aspekta procesa simulacije pomoću mrežnog simulatora NS2.

Široka rasprostranjenost Windows platforme uslovlila je

porast broja korisnika simulatora NS2 u Windows okruženju, poslednjih godina.



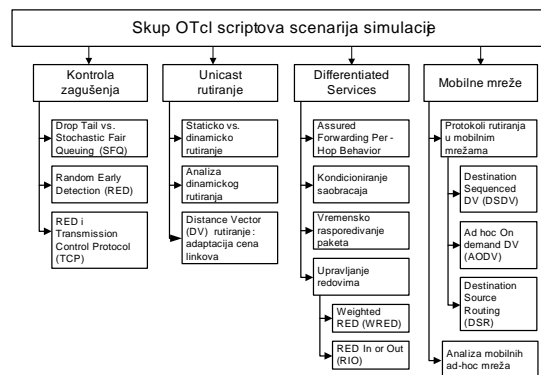
Sl.1. Korisnički aspekt simulacije pomoću NS2 [2]

Međutim, Linux pruža komforniji rad prilikom programiranja skriptova, prvenstveno zbog preglednog editora teksta.

C. Skup scenarija simulacije

Programirani su skriptovi za analizu kontrole zagušenja, kvaliteta servisa (diferencirani servisi – DiffServ), rutiranje (statičko i dinamičko) i MANET (slika 2). Posebna pažnja je posvećena podešavanju karakteristika čvorova za simulacije povezivanja fiksnih i mobilnih čvorova gde su od velike važnosti zadata međusobna rastojanja, parametri interfejsa, tip kanala i tip propagacije.

Analiza rezultata simulacija usredsređena je na mere performansi kao što su protok, vreme prenosa (kašnjenje), džiter, verovatnoća gubitka paketa. Takođe je analizirano preusmeravanje saobraćaja u slučajevima ispada linkova ili zagušenja (protok IP i kontrolnih paketa).



Sl.2. Skup OTcl skriptova simuliranih za potrebe analize

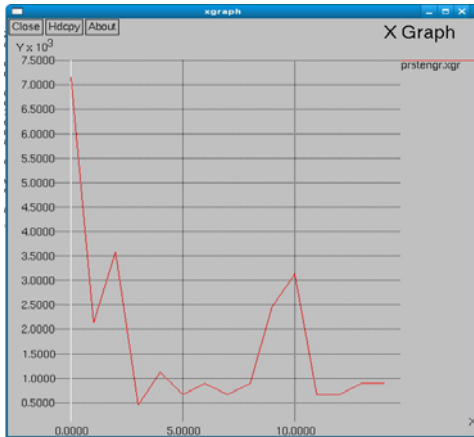
D. Alati za animaciju i analizu rezultata

Simulator je spregnut sa alatom za animaciju – NAM, koji omogućava vizuelno praćenje procesa simulacije – tokova saobraćaja, stanja linkova, stanja mrežnih čvorova, stanja redova i dr. Preko korisničkog interfejsa koji je sličan CD player-u moguće je pokrenuti, ubrzati, usporiti i zaustaviti simulaciju ili je pozicionirati na željeno mesto na vremenskoj osi. Rad NAM alata i grafički korisnički interfejsi su isti u Windows i Linux okruženju.

Analiza rezultata simulacije obavlja se na osnovu zapisa realnog saobraćaja u *trace* fajl koji generiše simulator. U *trace* fajl se upisuje unapred definisani skup događaja (generisanje paketa, prosleđivanje, odbacivanje paketa i

dr.) sortiran po diskretnom vremenu simulacije.

NS2 Linux verzija za potrebe grafičke interpretacije dobijenih rezultata koristi napredni Xgraph program koji dolazi u paketu sa NS2 instalacijom.



Sl.3. Izgled Xgraph korisničkog interfejsa

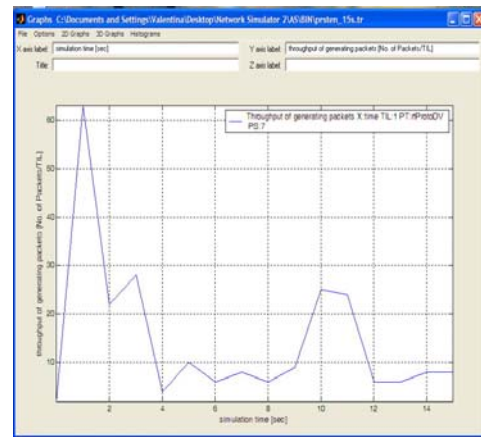
Rad Xgraph aplikacije se zasniva na očitavanju željenih podataka iz *trace* fajla u skladu sa prethodno generisanim skriptom `.awk` tipa. `Awk` je ekstenzija za skriptove kojima se definiše koje informacije iz *trace* fajla su potrebne i te informacije se izdvajaju na uređen način. Aplikacija Xgraph nije dostupna za Windows platforme.

Trace Graph je noviji alat, raspoloživ za Windows i Linux platforme, specifično razvijen za obradu i analizu NS2 *trace* fajlova.

Na slikama 3 i 4 prikazani su primeri Xgraph i Trace Graph korisničkog interfejsa, respektivno. Dijagrami predstavljaju protok kontrolnih paketa u simulaciji kojom je analizirano dinamičko rutiranje saobraćaja, u uslovima ispada mrežnog linka. U slučaju Xgraph aplikacije u Linux verziji grafički prikaz željenih podataka simulator generiše odmah nakon završetka simulacije, specijalnim „pop-up“ prozorom.

Za razliku od toga, Trace Graph mora posebno da se pokreće svaki put kada želimo da analiziramo rezultate. Xgraph nema mogućnost samostalnog izračunavanja parametara već isključivo zavisi od `awk` skripta koji je uključen u OTcl skript i poziva se u delu obrade *trace* fajla. Svaka naknadna promena mora biti prvo uneta u OTcl skript, zatim se pokreće simulacija i dobija se novi Xgraph prikaz. Tracegraph ima integrisan skup algoritama za izračunavanje različitih mera performansi.

Pri tome se ne zahteva ponovno pokretanje simulacije da bismo dobili neku dodatnu informaciju iz *trace* fajla. Dovoljno je u bogatom meniju pronaći opciju kojom se predstavlja mera performanse i ona će biti prikazana istog trenutka. Iako ovo svojstvo ubrzava proces analize rezultata, može se pojaviti problem ako se traži analiza koja nije obuhvaćena postojećim skupom algoritama.



Sl.4. Izgled Tracegraph korisničkog interfejsa

E. Poređenje scenarija simulacije

Obuhvaćeni scenariji (slika2) za DiffServ i fiksnu infrastrukturu pokazuju ujednačene rezultate u oba operativna sistema, s tim što postoje izvesne razlike u resursima koji se zauzimaju kao i vremenu potrebnom za izvođenje simulacija. U tabeli 1 je dat prikaz zauzetih resursa i vremena trajanja simulacija u zavisnosti od veličine simulirane mreže. Simulirana mreža se u početku sastoji od 6 čvorova povezanih dupleksnim linkovima u prsten, a veličinu postepeno povećavamo do 500 čvorova. Linkovi su kapaciteta 3 Mb/s sa kašnjenjem usled propagacije – 5 ms. Posmatra se DV (*Distance Vector*) dinamički protokol rutiranja. Simulirani su deterministički izvori saobraćaja CBR (*Constant Bit Rate*) povezani na agente UDP (*User Datagram Protocol*) transportnog protokola. Veličina CBR paketa je 500 bajtova a paketi se generišu u intervalu od 5ms. Simulirana su dva ispada linka u intervalima [1.0, 2.0] i [9.0, 10.0]. Trajanje svih simulacija je podešeno na 15 sekundi. Podaci pokazuju da je rad u Linux okruženju povoljniji, zauzima se manje slobodne memorije i simulacije traju kraće. Povećanje broja čvorova iznad 500 znatno usporava izvođenje simulacije. Linux je i u tom slučaju u prednosti.

TABELA 1: VREME IZVRŠAVANJA I POTREBNI MEMORIJSKI RESURSI

Br. čvorova	Windows	Linux
6	0:00:45, 4.7MB	0:00:02, 4.2 MB
100	0:20:00, 67.8 MB	0:01:40, 63.9 MB
300	0:45:00, 78.6 MB	0:15:10, 74 MB
500	5:44:00, 89.13MB	2:26:00, 84 MB

Dalji rad je bio usmeren na analizu MANET scenarija sa ciljem ispitivanja realnih mogućnosti simulatora u oba okruženja. Problem u Windows okruženju se javio kod pokušaja da se izvedu simulacije i analiziraju rezultati u slučaju integracije MANET sa fiksnom telekomunikacionom infrastrukturom. Zbog ograničenog prostora u nastavku će biti prikazani samo karakteristični scenariji.

```

# Definisanje parametara
set opt(chan) Channel/WirelessChannel
set opt(prop) Propagation/TwoRayGround
set opt(adhocRouting) DSDV
set opt(wirelessNodes) 102 ;#mob cvorovi
set opt(wiredNodes) 2 ;#hostovi
set opt(gatewayNodes) 4 ;#broj gejtveja
...
# Podešavanje parametara interfejsa
Phy/WirelessPhy set RXThresh_ 3.652e-10
...
# Kreiranje objekta simulatora
set ns[new Simulator]
$ns node-config -addressType hierarchical
....
# Podesavanje kanala, modela i topologije
set wtopo [new Topography]
# Kreiranje trace objekata za ns i nam
set tracefd [open $opt(tr) w]
set namtrace[open $opt(nam) w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace
$opt(x) $opt(y)
# Konfigurisanje cvorova
for {set i 0} {$i<$opt(nn)} {incr i} {
set node_($i) [$ns_ node]
$node($i) random-motion 0
...
$node($i) topography $wtopo
#Procedura za kreiranje topologije
Topologija
# Procedura za kreiranje izvora saobracaja
CBR
FTP
# Podesavanje vremena prekida simulacije
for {set i 0} {$i<$opt(nn)} {incr i} {
$ns_ at $opt(stop).000000001 "$node_($i)
reset"
...
$ns_ halt"
puts "Start simulacije..."
$ns_ run

```

Sl.5. Integracija MANET sa fiksnom infrastrukturom

Primer skripta dat je na slici 5, a pretpostavlja 102 mobilna čvora, 4 gejtveja i 2 hosta. Hijerarhijska topologija mreže sastoji se od dva domena sa dva, odnosno četiri poddomena, respektivno. Poddomeni imaju 1, 1, 28, 26, 26 i 26 čvorova, respektivno. Posmatra se ad hoc rutiranje DSDV protokolom. Fiksna infrastruktura (hostovi) od dva čvora je povezana sa gejtvejima dupleksnim linkovima, protoka 2 Mb/s, maksimalnog dozvoljenog kašnjenja 10 ms dok su dva žična čvora međusobno povezana dupleksnim linkom protoka 10Mb/s.

TABELA 2: REZIME ANALIZE

Karakteristika	Windows	Linux
Instalacija	Iz delova	„All-in-one“ ili iz delova
Animacija	NAM	NAM
Analiza rezultata	Tracegraph	Xgraph, Tracegraph
Rezultati simulacije	Sporiji; problem sa ad hoc mrežama, zauzima više resursa	Brže simulacije; ad hoc mreža, zauzima manje resursa
Skalabilnost	Do 30 čvorova (MANET)	Preko 100 čvorova

Simulirani su deterministički izvori saobraćaja povezani na agente UDP protokola, kao i FTP (*File Transfer Protocol*) izvori saobraćaja koji su povezani na TCP (*Transmission Control Protocol*) agente. Varijacijom broja čvorova utvrđeno je da su ovakve simulacije na Windows platformi mogle da se izvršavaju samo sa relativno malim brojem čvorova (do 30), a u Linux okruženju simuliramo bez problema preko 100 mobilnih čvorova. Rezime analize prikazan je u tabeli 2.

IV. ZAKLJUČAK

U radu je prikazano poređenje karakteristika simulatora NS2 na operativnim sistemima Windows i Linux. Skriptovi za fiksne mreže su bez većih problema simulirani na obe platforme i dobijeni su ekvivalentni rezultati. Za simulaciju većih mreža, pri radu u Linux okruženju zauzima se manje slobodne memorije i simulacije traju kraće. U slučaju scenarija sa MANET i, posebno, integracije MANET sa fiksnom telekomunikacionom infrastrukturom u Windows okruženju se pojavljuje problem loše skalabilnosti.

Prikazani rezultati predstavljaju osnov za sveobuhvatnu komparativnu analizu rada simulatora NS2 u Linux i Windows okruženju, prvenstveno u pogledu proširivanja skupa testiranih skriptova i ispitivanja skalabilnosti, efikasnosti i utroška računarskih resursa.

LITERATURA

- [1] S. Floyd, V. Paxson, "Difficulties in Simulating the Internet", *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, August 2001, pp. 392-403.
- [2] M. Stojanović, V. Aćimović-Raspopović, „Inženjering telekomunikacionog saobraćaja u multiservisnim IP mrežama“, naučna monografija, Saobraćajni fakultet u Beogradu, 2006.
- [3] G. F. Lucio et al., "OPNET Modeler and NS-2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed", *Proceedings of the International Conference WSEAS 2003*, Greece, October 2003.
- [4] D.M. Nicol, "Comparison of Network Simulators", Available: <http://www.ssfnet.org>.
- [5] "Network Simulator NS2 and Network Animator NAM". Available: <http://www.isi.edu/nsnam>.
- [6] J. Malek, "Trace Graph – Network Simulator NS Trace Files Analyzer", 2003. Available: <http://www.tracegraph.com/>

ABSTRACT

This paper presents a comparative study of behaviour of Network Simulator version 2 (NS2) in Linux and Windows operating system environments. The experimental comparison relies on programming a set of OTcl scripts and testing simulator performance on both operating systems. The analysis considers the following aspects: installation, user aspect, animation, analysis tools, comparison of simulation scenarios and scalability.

ANALYSIS OF NS2 SIMULATOR FEATURES ON LINUX AND WINDOWS OS PLATFORMS

Valentina Timčenko, Slavica Boštjančič Rakas,
Mirjana Stojanović