

# Analysis and importance of using simulators in data network

Adrian Mihai GAITAN, Valentin POPA, Vasile Gheorghiță GAITAN

**Abstract**—A methodology for analyzing large-scale performance in engineering systems is the simulation of network communication. In many areas of engineering and science, a computer can be used to simulate a natural or artificial phenomenon instead of experimenting it, using a real system. Examples of such experiments made on computers are the simulation studies of a network congestion control and competition for resources in a computer operating system. The objective of this article is to classify and compare network simulators to select the closest simulation tools necessary for the research process.

**Index Terms**—Network Simulators, OPNET Modler, Ns-2

## I. INTRODUCERE

**S**IMULAREA este un experiment ce determină caracteristicile unui system din punct de vedere empiric. Este o metodă de modelare care imită sau permite imitarea unui comportament al unui sistem în timp. Aceasta implică generarea și observarea istoricului artificial al sistemului studiat, ce duce la concluzii în legătură cu comportamentul dinamic al sistemului real. În ciuda unor dificultăți, simularea este larg folosită în practică și utilizarea simulării va crește sigur, întrucât experimentarea cu sisteme reale este din ce în ce mai dificilă datorită costului său și a altor motive. Simularea este una dintre cele mai populare metode de a analiza sistemele bazate pe stare. Este larg aplicată simularea evenimentelor discrete (DES) [1][2] pentru studierea unor sisteme cu cozi de așteptare și a unor rețele aleatoare Petri. De aici importanța ca orice inginer de calculatoare să fie familiarizat cu bazele simulării. Metodologia de simulare a devenit populară în rândul cercetătorilor din domeniul rețelelor de calculatoare și telecomunicații cât și în rândul dezvoltatorilor la nivel mondial. Această popularitate se datorează unei varietăți de pachete de simulare sofisticate și puternice și de asemenea flexibilității în construirea și validarea unui model oferit de simulare [3][4]. Pentru selectarea unui simulator de rețea adecvat, pentru o sarcină de simulare, este important să avem o bună cunoaștere a instrumentelor de simulare disponibile, ținând cont de punctele lor forte cât și de cele slabe. De asemenea trebuie să se țină cont ca rezultatele obținute din simulare să fie valide și credibile.

## II. CLASIFICAREA INSTRUMENTELOR ÎN DES

Instrumentele de simulare pot fi împărțite în patru categorii de bază (Fig.1.):

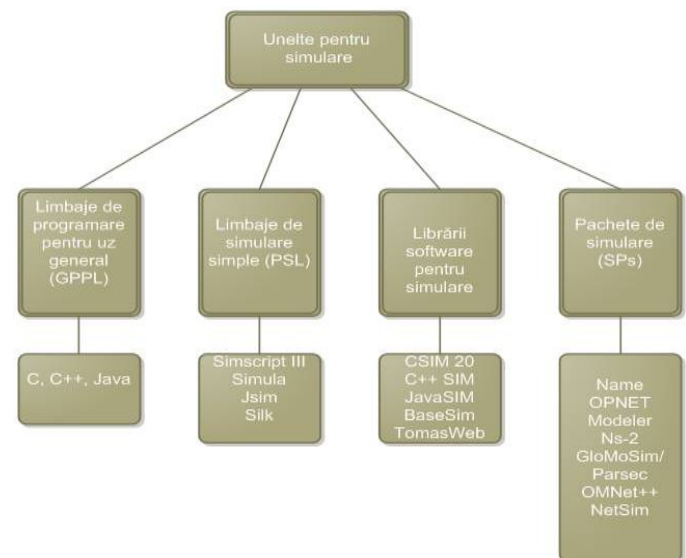


Fig.1. Clasificarea instrumentelor de simulare.

### A. Limbaje de programare de uz general (GPPL)

C, C++ , Java sunt limbajele care au avantajul de a fi foarte disponibile. Acestea oferă un control total asupra unui proces de dezvoltare al unui soft. Dezavantajul este că pentru a construi modelul este nevoie de mult timp. De asemenea, nu oferă nici un suport de bază pentru DES. Generarea de variabile aleatoare pentru diferite distribuții aleatoare și metoda statistică a rezultatului trebuie învățată și programată.

### B. Limbaje de programare simple (PSL)

Simula, Simscrip III, Jsim, Silk sunt doar câteva exemple. Un dezavantaj ar fi că nu sunt disponibile imediat. E nevoie și de cunoștințe de programare în noul limbaj înainte de execuția modelelor de simulare.

### C. Librării software pentru simulare

CSIM-20, C++SIM, și echivalentul său JavaSim, TomasWeb, sunt câteva exemple. Ele oferă o colecție extinsă scrisă în GPPL (C, C++, Java) care ajută la dezvoltarea și execuția modelului de simulare. Oferă funcții variate de

colectare de date, de monitorizare și statistice, fără a fi necesar efortul de a învăța un limbaj nou.

#### D. Pachete de simulare

OPNET Modeler **Error! Reference source not found.**, Ns-2 [5], Qualnet [7], sunt câteva exemple. Ele facilitează procesul de construire a unui model, majoritatea dintre ele oferind interfețe grafice pentru utilizator împreună cu flexibilitatea lucrului la un nivel textual. Oferă un suport de bază pentru DES și analiză statistică. Acoperă de asemenea mai multe domenii de aplicare precum rețelele TCP/IP. Asigură suportul ca timpul de construire a unui model să fie mai scurt. Anumite instrumente de simulare ca OPNET Modeler oferă de asemenea utilizatorului o opțiune pentru a selecta soluții analitice ale modelului. Ca și PSL, SP necesită cunoștințe în noul limbaj /mediu și tind să fie mai puțin flexibile decât PSL.

### III. PACHETE DE SIMULARE

În timp ce există diferite simulatoare pentru a construi o varietate de modele de rețea, vor fi comparate două dintre cele mai populare simulatoare de rețea evidențiind punctele lor forte și punctele slabe. Aceste simulatoare au fost selectate pe baza popularității lor, rezultatelor publicate și caracteristicilor interesante.

#### A. OPNET : Optimized Network Engineering Tool

Este un mediu de simulare comercial cu suport GUI (dezvoltat inițial de MIT). A fost proiectat pentru a suporta modelarea, proiectarea și validarea arhitecturilor și protocoalelor de comunicație. Este folosit pentru planificarea rețelelor (LAN/WAN). Are librării incluse pentru toate standardele protocolalelor TCP/IP și aplicații incluzând IP Quality of Service (QoS), Resource Reservation Protocol (RSVP), etc. Poate gestiona rețele bazate pe fibră optică și microunde. Evaluează algoritmi noi de rutare pentru rutere, comutatoare (switchuri) și alte dispozitive de conectare, înainte ca acestea să fie instalate fizic în rețea. OPNET furnizează pentru construirea modelului o abordare orientată pe obiect. Interfața grafică utilizator bazată pe meniuri ajută utilizatorul la proiectarea modelului. OPNET Modeler oferă opțiuni pentru cele mai importante șabloane privind traficul de la o sursă externă cum ar fi un fișier XML, spreadsheet sau un server OPNET VNE. OPNET Modler permite utilizatorului să modeleze topologii de rețea utilizând trei niveluri ierarhice [8][9]:

Nivelul rețea: este nivelul cel mai de sus al modelării în OPNET Modler. Topologiile sunt modelate utilizând componente de la nivelul rețea precum rutere, gazde și legături. Aceste modele de rețele pot fi aduse și plasate dintr-o paletă de obiecte, pot fi alese din meniul OPNET care conține numeroase topologii ca stea, inel, plasă sau pot fi importate de la o rețea reală prin colectarea de informații privind topologia rețelei.

Nivelul nod: acest nivel modelează structura internă a componentelor de la nivelul rețea. Se capturează arhitectura unui dispozitiv din rețea sau sistem prin descrierea interacțiunilor dintre elementele funcționale denumite module. Modulele au capacitatea de a genera, trimite și recepționa pachete de la alte module pentru a realiza funcțiile lor în cadrul nodului. Modulele sunt conectate de șiruri care pot fi șiruri de tip pachet, șiruri statistice sau un șir asociat. După cum sugerează numele șirurile de tip pachet reprezintă fluxul de pachete dintre module; șirul statistic este utilizat pentru a transporta statistica unui șir între module. Un șir asociat este utilizat pentru asocierea logică a diferitelor module și nu transportă nici o informație.

Nivelul proces: acest nivel utilizează descrierea automatelor cu stări finite (FSM) pentru a suporta specificațiile protocoalelor, resursele, aplicațiile, algoritmi și politicile aplicate coșilor. Stările și tranzițiile definesc grafic evoluția procesului ca răspuns la un eveniment. Fiecare stare a modelului de proces conține cod C/C++, suportat de o librărie extinsă pentru protocoalele de comunicație.

#### B. Ns-2: Network Simulator

Este unul din cele mai utilizate simulatoare de rețea utilizate în prezent pe scară largă. Este un simulator de evenimente discrete pentru aplicații în rețea. Ns a început ca variantă a simulatorului de rețea REAL cu suportul DARPA (Defense Advanced Research Project Agency) și a unor universități cum ar fi Universitatea Berkley din California sau Universitatea Carnegie Mellon din Pittsburgh. Versiunea sa curentă este scrisă în C++ și TCL orientat pe obiect. Ns-2 este un interpretor de scripturi OTcl orientate pe obiect care conține un planificator de simulare a evenimentelor, componente de rețea, librării cu obiecte și librării cu module de setare a rețelei [10]. Facilitățile oferite de Ns-2 constau în furnizarea de sub-modele ambalate pentru câteva protocoale de rețea cum ar fi TCP și UDP, mecanisme de gestiune a coșilor din rutere cum ar fi tail-drop, RED, algoritmi de rutare CBQ precum Dijkstra [11]. Furnizează câteva surse de trafic cum ar fi Telnet, FTP sau CBR. Implementează câteva protocoale MAC pentru LAN și protocoale multicast. Conține un simulator de evenimente și un număr larg de obiecte de rețea, cum ar fi ruterele, legăturile, etc, care sunt interconectate pentru a forma o rețea. Utilizatorul trebuie să scrie un script Otcl care inițiază un planificator de evenimente, setează topologia rețelei utilizând obiecte de tip rețea, și spune sursei de trafic când să pornească sau să oprească transmiterea pachetelor prin planificatorul de evenimente. Toate componentele și caracteristicile rețelelor sunt reprezentate de către clase [12]. După cum se poate vedea în Figura 2 toate obiectele sunt derivate din clasa NsObject. Aceasta este alcătuită din două subclase: conectori și clasificatori. Conectorul este un obiect NsObject de la care sunt derivate legăturile precum coșile și întârzierile. Clasificatorii examinează pachetele și le trimit mai departe

către destinația corespunzătoare.

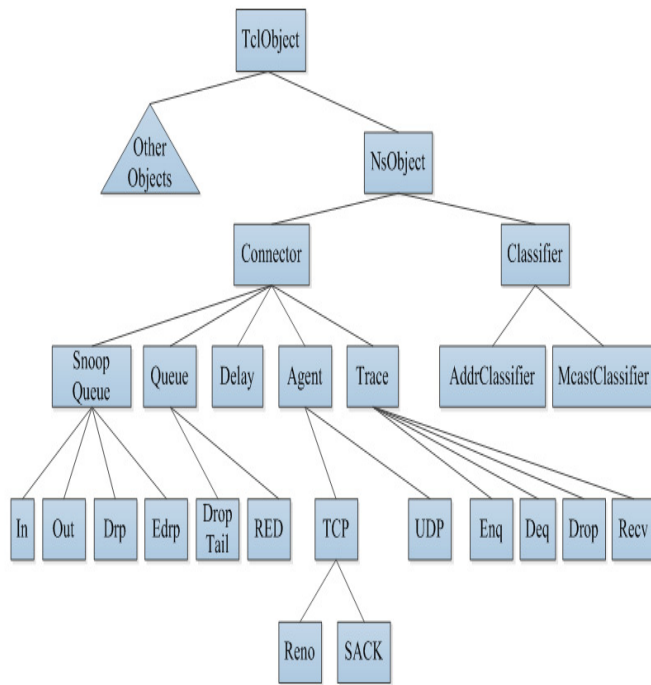


Fig.2. Ierarhia preluată din NS prin exemple [12].

Cele mai utilizate obiecte în Ns-2 sunt:

**Noduri:** acestea reprezintă clienți, gazde, rutere și comutatoare. Pot fi definite două tipuri de noduri și anume unicast și multicast.

**Clasificatori:** aceștia determină obiectul pentru interfațarea cu ieșirea pe baza adresei sursă și a adresei destinație a pachetului. Exemple de clasificatori sunt clasificatori de adresă, multicast, multicale și replicatori.

**Legături:** acestea sunt utilizate pentru conectarea nodurilor pentru a forma o topologie de rețea. O legătură este definită prin capătul ei care devine punct de intrare, o referință la elementul principal al cozii și coadă pentru a procesa pachetele rejectate pe legătură. Activitățile pe rețea sunt trasate în jurul legăturilor simple.

**Agenți:** aceștia sunt puncte finale de transport de unde pleacă sau ajung pachetele. Tipuri de agenți sunt TCP și UDP. Ns-2 suportă un număr mare de variante TCP și oferă opțiuni pentru a seta bitul ECN, mecanismul de control al congestiei, și setarea ferestrei.

**Aplicații:** tipurile principale de aplicații pe care le suportă ns-2 sunt generatoarele de trafic și simularea aplicațiilor. Unele di aplicațiile bazate pe TCP suportate de Ns-2 sunt Telnet și FTP.

**Generatoarele de trafic:** în cazul unei simulări bazate pe distribuții este necesară generarea automată de trafic cu forma și șablonul dorit. Unele din generatoarele de trafic pe care le furnizează Ns-2 sunt: Poisson, On-Off, CBR și Pareto On-Off.

În Tabelul I sunt prezentate câteva din avantajele și

dezavantajele folosirii simulatoarelor OPNET Modler și Ns-2 în domeniul rețelelor de calculatoare.

|             | OPNET Modler                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Ns-2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Avantaje    | <ul style="list-style-type: none"> <li>• oferă facilități de animație care pot ajuta în înțelegerea și depanarea rețelei;</li> <li>• modelul de dezvoltare modular;</li> <li>• un nivel ridicat de detaliere a modelării;</li> <li>• rulează pe diferite platforme precum Windows, Linux și Unix;</li> <li>• interfața prietenoasă cu utilizatorul (GUI);</li> <li>• este posibilă și simularea paralelă;</li> <li>• funcție internă pentru calculul intervalului de încredere.</li> </ul> | <ul style="list-style-type: none"> <li>• disponibilitatea sa sub licență publică;</li> <li>• poate produce simulări lungi, dar foarte precise, ale caror rezultate pot fi mai ușor acceptate de comunitatea științifică;</li> <li>• implementează protocoale reale (ex. TCP, IP, UDP, etc) într-un format intern, modelul de simulare urmând să "asambleze" protocoalele respective;</li> <li>• rulează pe câteva platforme precum Unix, Linux, și Windows (versiune Cygwin).</li> </ul> |
| Dezavantaje | <ul style="list-style-type: none"> <li>• este un pachet de simulare foarte scump (costul licenței și al întreținerii fiind ridicate);</li> <li>• locul său în categorie nu este unul foarte transparent.</li> </ul>                                                                                                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>• ns-2 nu oferă o consistență privind codul și documentația la toate variantele;</li> <li>• spre deosebire de OPNET Modler, ns-2 nu are o funcție internă pentru calculul intervalului de încredere.</li> </ul>                                                                                                                                                                                                                                   |

Tabelul I: Comparare OPNET Modler vs. Ns-2

#### IV. CONCLUZII

În concluzie, simulările ocupă un loc foarte important în estimarea diferitelor măsuratori din cadrul unui sistem. Este larg aplicată simularea evenimentelor discrete (DES) pentru studierea unor sisteme cu cozi de așteptare și a unor rețele aleatoare Petri. DES utilizează ca intrări numere aleatoare astfel încât să se producă diferite seturi de ieșiri, în general pentru fiecare execuție. Analiza ieșirilor utilizează tehnici statistice care folosesc principiul deducției pentru a analiza seturilor de date. Simularea poate fi utilizată în evaluarea performanțelor, a siguranței și a disponibilității sistemelor. În această lucrare au fost prezentate și analizate avantajele și dezavantajele a două pachete de simulare OPNET Modler și Ns-2. Au fost prezentate câteva din caracteristicile esențiale ale acestora în cadrul domeniului rețelelor de calculatoare. Simulatoarele facilitează și accelerează procesul de construire a unui model, majoritatea dintre ele oferind utilizatorului interfețe grafice intuitive integrând flexibilitatea posibilității de lucru și la nivel textual. Un alt avantaj major îl reprezintă suportul pe care îl oferă pentru analiza DES și elemente statistice.

## V. MULȚUMIRI

Această lucrare a beneficiat de suport financiar prin proiectul "Provocările cunoașterii și dezvoltare prin cercetare doctorală PRO-DOCT Contract nr.POSDRU/88/1.5/S/52946", proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.

## BIBLIOGRAFIE

- [1] J. J.Banks, J.S. Carson, B.L. Nelson, D.M. Nicol, *Discrete-Event System Simulation*, Prentice Hall, Upper Saddle River, NJ, third edition, 2001.
- [2] A.M . Law, W.D. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill Higher Education, third edition, 2000.
- [3] K. Pawlikowski, H.-D. J. Jeong, and J.-S. R. Lee, "On credibility of simulation studies of telecommunication networks," *IEEE Communications Magazine*, vol. 40, no. 1, pp. 132-139, 2002.
- [4] B. Schmeiser, "Simulation output analysis: A tutorial based on one research thread," presented at the 2004 Winter Simulation Conference, December 5-8, 2004, pp. 162-170.
- [5] OPNET Technologies. Retrieved February 10, 2011, Available : [www.opnet.com](http://www.opnet.com)
- [6] Network Simulator Available at <http://www.isi.edu/nsnam/ns>.
- [7] Scalable Network Technologies. QualNet Developer. Retrieved March 5, 2011, Available: <http://www.qualnet.com/products/developer.php>
- [8] M.Hassan, R .Y. Jain, *High Performance TCP/IP Networking: Concepts, Issues, and Solutions*, Prentice-Hall 2003.
- [9] X. Chang, "Network simulation with opnet," presented at 1999 winter simulation conference, Phoenix, AZ, USA, 1999, pp. 307-314.
- [10] K. Fall and K. Varadhan. The ns manual. The VINT project. Retrieved February 10, 2011, Available at <http://www.isi.edu/nsnam/ns/doc/>.
- [11] E.W. Dijkstra, A Note on Two Problems in Connection with Graphs, *Numerische Math* 1, pp.269-271, 1959.
- [12] J .Cheung, M.Claypool, *NS by example*, Available <http://nile.wpi.edu/NS/>.