

# Az IP hálózati forgalomgenerálás és alkalmazásai

## **Absztrakt**

Az IP hálózatok pontos méretezéséhez és a szűk keresztmetszetek megtalálásához nélkülözhetetlen eszközök a forgalomgenerátorok. Cikkünkben áttekintjük a jelenleg használatos forgalomgenerátor rendszereket és bemutatunk egy újszerű megoldást.

## **1 A forgalomgenerálás módszerei**

Az Internet forgalomelméleti kutatásainak egyik fő célja a hálózati alkalmazások által generált forgalom tulajdonságainak feltárása. Ezen ismeretek alapjánvan lehetőség hatékony tervezési eljárások kidolgozására. Az új protokollok, hálózati eszközök és alkalmazások teljesítménytesztelése is további fontos feladatok. Ezen célok elérésére két fő megközelítés létezik, a szimulációs módszerek és a hálózatokon történő mérés és analízis forgalomgenerátorok segítségével.

A szimuláció során egyetlen gépen szoftveres programcsomag szimulálja akár sok száz hálózati eszköz, számítógép működését. Nagyon sok sikeres szimulációs programcsomag létezik, például a REAL [1] és az NS [2], melyeket igen nagy sikerrel alkalmaznak. Hatalmas előnyük a kis hardver igény és – főleg gyorsított szimuláció esetén – a viszonylag rövid időszükséglet. Használatuk mégis csak elnagyolt, esetleg alapvetően téves következtetésekre vezethet, ugyanis csak a valós világ absztrakcióján dolgoznak. A hálózati eszközök, a számítógépek, az operációs rendszerek nagyon bonyolult, összetett és sokszor véletlenszerű működéséről keveset tudunk, ráadásul a különböző környezeti tényezők és egyéb, a nyílt hálózatban levő gépek hatását nagyon nehéz figyelembe venni. Ezért a szimulációs környezet többnyire egy erősen leszűkített, az eszközök és szoftverek specifikációját követő absztrakció, amely sok esetben nagyon távol áll a valós működéstől. E hiányosság kiküszöbölése a forgalomgeneráló és mérő rendszerek kifejlesztésének fő motiválója.

A forgalomgeneráló és mérő rendszerek olyan, többnyire elosztott szoftver rendszerek, amelyek sok hálózati eszköz, számítógép között a valós hálózaton generálnak forgalmat és ezen forgalom a hálózat különböző pontján elvégzett méréseiből vonják le a következtetéseket. Ezen eszközök sokrétűen felhasználhatóak az egyszerű teljesítményteszteléstől, a két végpont közti áteresztőképesség és sávszélesség méréstől kezdve a protokolltesztelésig és hálózatoméretezésig. Nagy előnyük, hogy valós környezetben mérnek, viszont a pontos és megbízható méréshez viszonylag nagy hardverigény és jelentős időszükséglet párosul, mivel statisztikailag korrekt eredményekhez nagy számú mérési eredmény szükséges.

A forgalomgenerátor rendszerek fejlesztése napjainkban is folyik. Ezek az eszközök általában csak előre definiált csomagokat küldenek át állandó vagy véletlenszerűen változó várakozási időket beiktatva és esetleg néhány egyszerűbb, csak néhány paraméterben változtatható alkalmazásmodellt is tartalmaznak.

A passzív mérőeszközökkel a valós felhasználók által a hálózaton generált forgalmat vizsgálhatjuk. Legelterjedtebben a két csomópont közé iktatható monitorozó rendszereket használnak. Mivel ezek az eszközök csak detektálják az adott pontban a

hálózaton keresztülhaladó adatfolyamot, így nem avatkoznak bele a rendszerbe. Passzív mérőrendszerek segítségével különféle forgalmi karakterisztikákat, például csomagméret eloszlást, érkezési időközöket, vagy a forgalom összetételét (TCP, UDP, stb.) mérhetünk. Legismertebb az NLANR projektje által használt OCXmon [3] monitorozó rendszer, melynél két mérő kártyát (mindkét irányban lehet vizsgálni a forgalmat) helyeznek egy nagyteljesítményű számítógépbe, és ehhez csatlakoztatják az optikai kábelt. A vezérlő szoftver nagy előnyei, hogy teljes körű felhasználhatóságot és nagy mennyiségű tesztelési információt biztosít. Ezen rendszerek hátránya a nagy költségigény, a mobilitás hiánya, valamint a korlátozott lehetőségek. A költségeket növeli, hogy valamennyi számítógép külön hardware-t igényel, a lehetőségeket pedig az korlátozza, hogy statisztikai megfontolások miatt csak a meglévő, nagy forgalmú helyeken érdemes használni az eszközt, és teszt alatt begyűjthető adatmennyiség is korlátozott.

Aktív mérőeszközöknek is nevezhetnénk a legegyszerűbb forgalomgenerátorokat, mivel ezek elsődleges célja nem a hálózati terhelés létrehozása, hanem különböző hálózati paraméterek mérése. Többnyire olyan ICMP vagy UDP alapú és hálózatmenedzselési célú programokból származnak, mint a Ping vagy a TraceRoute, s az aktív csomaggenerálás a működésük velejárója, nem a célja. Ilyen programok például a Pittsburgh Supercomputing Center-ben fejlesztett Poisson Ping (POIP), Traceroute Reno (TReno) [4], és a NIMI [5], valamint a PingER [6], a Pathchar [7], és a Nettimer [8], melyek két végpont között képesek mérni olyan paramétereket, mint késleltetés, körülfordulási idő, stb., illetve a már elosztott hálózatban több végpont között futtatható Skitter [9].

A már tényleges forgalomgenerátorok közül főleg két hálózati végpont közötti áteresztőképesség, és egyéb teljesítményjellemzők mérésre alkalmasak például a TTCP [10], valamint Hewlett-Packard számítógép-hálózati kutatólaboratóriumaiban fejlesztett NetPerf [11]. A SPECweb96 [12][13] specifikusan a Web-szerverek maximális áteresztőképességének meghatározására lett kifejlesztve. A Scalable URL Reference Generator (SURGE) [14], bár fő célja szintén csak HTTP szerverek teljesítménytesztelése, mégis fontos eszköz, mert igen fejlett felhasználó szintű beépített HTTP modellt alkalmaz, amely lehetővé teszi számára a valóságot megközelítő, bár igen statikus Web forgalom generálását. Az Internet Protocol Benchmark (IPB) [15] bármely két hoszt között futtatható, különböző alkalmazások teljesítménytesztelésére. Az IPB kliens és szerver oldali futtatható állományokból áll, és HTTP kérés-válasz alapon történik a Webes applikációk hálózati forgalmának generálása. A forgalmi modelleket hálózati forgalmak analíziséből merítették. Az eszköz már megközelítőleg jól utánozza a valós hálózati forgalmat. A japán Nara Institute-on kifejlesztett Distributed Benchmark System (DBS) [16] és a Kansas Egyetemen létrehozott NetSpec [17] generátorok már elosztott rendszerek, azaz képesek több gép között változatos forgalmat generálni, s ennek méréséből különféle statisztikákat készíteni. A DBS előre definiált csomagcsoportokat képes adott időközönként küldeni, míg a Netspec három beépített, igen egyszerű felhasználói szintű alkalmazásmodellt is tartalmaz. (TELNET, FTP, VIDEO)

## **2 Az Agent-Ant forgalomgenerátor rendszer**

A két utóbbi eszköz már igen fejlett és sokrétűen felhasználható, de mégis igen statikusak, nem képesek az Interneten rohamosan megjelenő új alkalmazások által

előállított forgalmat élethűen legenerálni, márpedig a pontos és megbízható hálózattervezés alapvető feltétele, hogy helyes modelljeink és jó generáló eszközeink legyenek kéznél. Sajnos helyes alkalmazásmodellek alkotása nem egyszerű feladat. Az elméleti eredményeket a gyakorlatban a generátorrendszerben kipróbálva, a generált forgalmat a valós hálózaton mért forgalommal összevetve, és a modelleket folyamatosan finomítva lehet eljutni egy jól használható, megközelítően valóságghű modellhez.

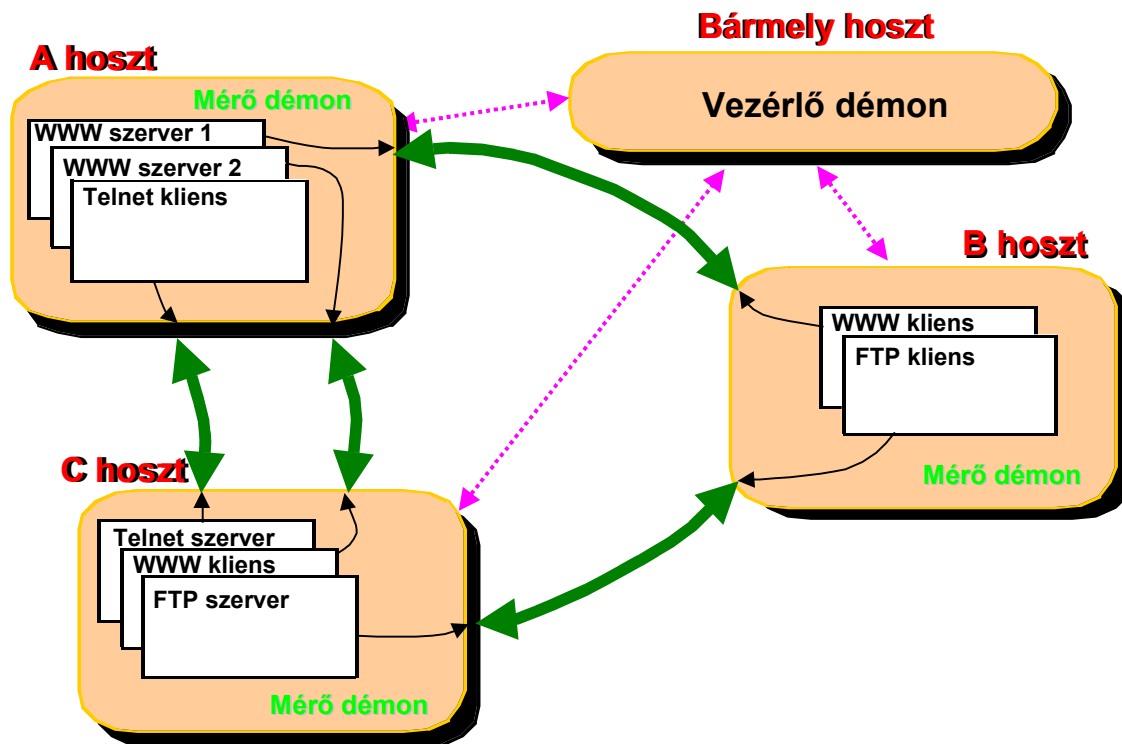
Célunk egy ilyen, bonyolult alkalmazásmodelleket kezelni képes elosztott generátorrendszer kifejlesztése, mely rendszer jelenleg már elérte a működőképes állapotot és néhány mérésorozatot is végeztünk vele. Munkánkat az 1998-ban öt ipari tag és három egyetemi tanszék együttműködésével megalakult Egyetemi Távközlési és Informatikai Központ támogatásával végezzük.

Az általunk fejlesztett forgalomgenerátor rendszer alapvető architektúrájában hasonló az elosztott DBS rendszerhez, viszont az eszköz képes bonyolult kérés-válasz alapú alkalmazásmodelleket kezelni, ugyanis beépített script nyelv - Tool Command Language (TCL) - interpretert tartalmaz, ami lehetővé teszi, hogy e magas szintű nyelven írt, könnyen fejleszthető és gyorsan módosítható modelleket használjon a forgalom generálására. Így lehetővé válik, hogy a lehető legjobban utánózzuk a valódi alkalmazások működését és a hálózaton megjelenő valós forgalmat a lehető legjobban közelítsük. A TCL-ben írt modellek függetlenek magától a rendszertől, így nincs szükség az eszköz újrafordítására, ha módosul a meglévő modell, vagy új modellel bővül az eszköz, tehát a modellek nagyon gyorsan fejleszthetők.

Az eszköz elnevezése Advanced GENERation Tool of Applications' Network Traffic, röviden Agent-Ant, és a továbbiakban ismertetjük a felépítését és a működését.

Az Agent-Ant két részre osztható, az alrendszerre, mely futtató környezetül szolgál a TCL nyelvű modelleknek, és magukra a modellekre. Az alrendszer további két részre osztható, a C nyelvű keretrendszerre és a TCL interpreterre. Az alrendszer felfogható, mint egy programozási interfész a modellek felé, tulajdonképpen felülről egy speciális funkciókkal kibővített TCL interpreternek látszik. A keretrendszer az Agent-Ant gerince. Ez a tulajdonképpeni elosztott architektúra, mely kialakítja a mérési környezetet, vezérli a működést és futtatja az interpretereken keresztül a modelleket.

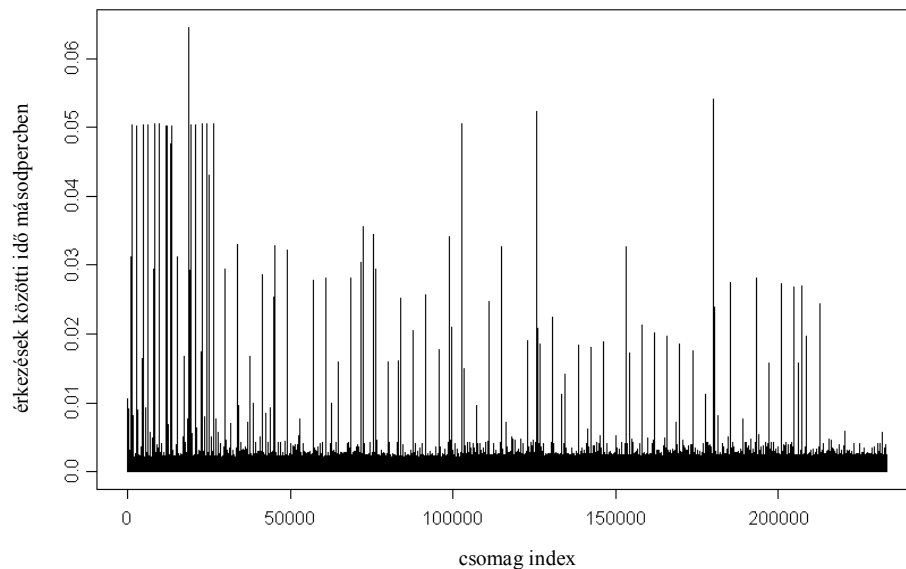
Maga a keretrendszer egy vezérlő és mérő démonokból áll. A vezérlő démon kommunikál a felhasználóval, aki egy konfigurációs fájlban megadja, hogy mely állomásokon milyen kliensmodelleket és milyen időzítéssel kíván futtatni, valamint magukat a modellscripthekeket is biztosítja. E tevékenységen túl a felhasználónak már csak egy-egy mérődemont kell indítania a mérésben résztvevő állomásokon. A felhasználó által a vezérlőnek megadott konfigurációs adatokból már maga a vezérlő számolja ki, hogy mely állomásokon, milyen szervermodelleket kell indítania. Ezután kontrol kapcsolatokat nyit a mérő démonok felé, melyek megsokszorozzák magukat, hogy megfelelő számú modell indulhasson. A vezérlő letölti a megfelelő szerver vagy kliens scripthekeket, melyeket a mérő démonok el is indítanak. Ezután már a kliens és szerver modellek tényleges egymás közti kommunikációja maga a forgalomgenerálás. A kommunikáció valós, például a kliens modell akár egy kis scriptrészletet is átküldhet, melyet a szerver lefuttat. Az előbbieken leírt architektúrát és működési elvet szemlélteti az 1. ábra.



**Ábra 1 Az Agent-Ant architektúrája**

Az ábrán a vékony szaggatott nyilak a vezérlő kapcsolatait mutatják, melyen a mérés elején a konfigurációt letölti a mérő démonoknak, és a végén pedig az eredményt gyűjti be. A vastag nyilak a modellek közti kétirányú kérés-válasz kapcsolatokat jelölik – a tényleges generált forgalmat. Mivel a modellek az alkalmazási szinten működnek és a kommunikációjukhoz az operációs rendszer nyújtotta összeköttetésalapú Transmission Control Protocol (TCP), illetve összeköttetésmentes User Datagram Protocol (UDP) IP hálózati szállítási protokollokat használják, ugyanazt a környezetet látják és használják, mint egy átlagos Internetes alkalmazás, például egy Web böngésző. Ezáltal tehát lehetővé válik, hogy maximálisan szimuláljuk az ilyen alkalmazások üzenetküldési viselkedését, s az operációs rendszer és a TCP/UDP protokollok működéséből adódó torzulások ugyanúgy megjelennek a generált forgalomban, mint a valós hálózaton mért IP forgalomban.

Erre a jelenségre már egy igen egyszerű mérési eredmény is rámutat. Kísérletünkben 5 megabájt hosszúságú üzeneteket küldtünk TCP kapcsolaton folyamatosan két számítógép között az Agent-Ant használatával. A külső környezetet teljesen kizártuk a mérés időtartamára. Az operációs rendszer és a protokollok természetesen apróbb csomagokra tördelve küldik el az üzeneteket. A hálózaton megjelent csomagok egymás utáni időközeit mutatja a 2. ábra.



**Ábra 2 Az érkezési időközök folyamatosan küldött üzenet csomagjai közt**

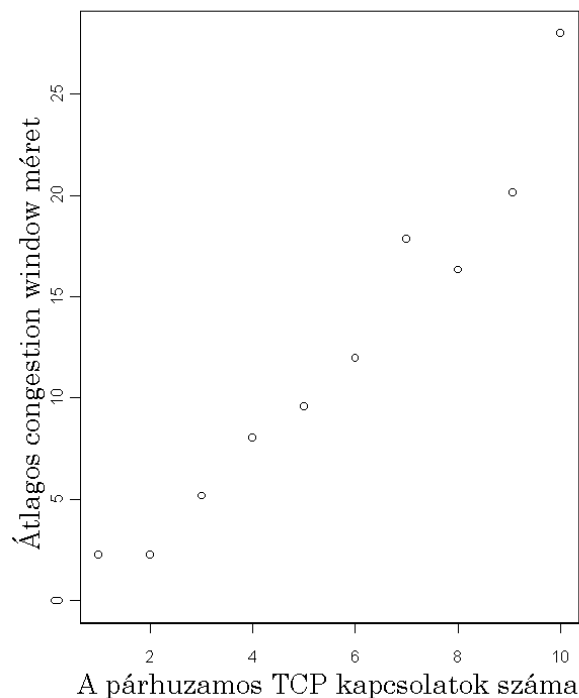
Az ábrán látható, hogy csomagok nem egyenletesen érkeznek, hanem némelyek kisebb, megint mások igen nagy késleltetést szenvednek. Valójában körülbelül háromszintű késleltetés figyelhető meg. A normál, egy-két milliszekundumos késleltetések, aztán az operációs rendszer tárolóinak működéséből és a TCP protokoll működéséből adódó, szinte periodikus húsz-harminc milliszekundumos késleltetések, majd a már abszolút nem periodikus majd száz milliszekundumos késleltetések, melyek valószínűleg a processzor időosztási mechanizmusa miatt következnek be.

Ezek a torzító hatások tehát az általunk alkalmazásszinten generált forgalomra is hatnak, s egyből egy fontos kérdést is felvetnek. Nevezetesen a szakirodalomban található modellek szinte mind a valós hálózaton mért csomagokból származtatott csomagszintű modellek, s így természetesen tartalmazzák e torzító hatásokat. Számunkra viszont fontos az alkalmazások által generált forgalom leírása, amely jórészt független a környezettől és a felhasználó befolyása alatt áll. Az Agent-Ant alkalmas e torzító hatások vizsgálatára is.

Bár az eszköz még fejlesztés alatt áll, máris bizonyította a használhatóságát. Legutóbbi eredményeinket a Budapesti hálózaton mérve kaptuk. Egyik vizsgálatunkban egy vállalati intranetről küldtünk 1500 bájtos UDP csomagokat a Budapesti Műszaki és Gazdaságtudományi Egyetem egyik gépére. A forgalom egy Matávnet útvonalválasztón keresztül zajlott és a csomagvesztések karakterisztikáját figyeltük különböző sávszélességű UDP folyamatok esetén. Noha jól ismert, hogy a csomagvesztések többnyire csoportosan történnek, mi mégis többnyire egyedi veszteségeket tapasztaltuk, csoportosakat ritkán. A mért csomagvesztések láthatóan nem-stacioner folyamatot követtek, és a vizsgálatok bizonyították, hogy függetlenek voltak egymástól. A csomagvesztési időközök vizsgálata során azt találtuk, hogy az legjobban az exponenciális eloszlással modellezhető. Ezekből az eredményekből azt a következtetést vontuk le, hogy a Matáv

útvonalválasztón a csomagvesztéseket modellezhetjük egy Poisson folyamat segítségével..

Másik vizsgálatunkban a párhuzamos TCP kapcsolatok számának növekedése és a körülfordulási idő (round-trip time - RTT), illetve a torlódási ablak (congestion window - cwnd) méretének kapcsolatát vizsgáltuk. A torlódási ablak a hálózaton utazó nyugtázatlan csomagok számát jelzi, s mivel ezek valamelyik hálózati komponensben várakoznak, befolyásolják a körülfordulási idő hosszát az útvonalon. Elméletileg a párhuzamos TCP kapcsolatok számának növekedésével a körülfordulási idő és a torlódási ablak is várhatóan lineárisan nő. Méréseink során azt tapasztaltuk, hogy a torlódási ablak valóban lineárisan nő a kapcsolatok számának növekedésével, amint az a Ábra 3. ábrán látható.



**Ábra 3 A torlódási ablak és a TCP folyamatok számának kapcsolata**

A körülfordulási idő viszont szakaszosan lineárisan nő. Az egyes szakaszokon lineáris a kapcsolat, de a párhuzamos kapcsolatok számának növekedésével egyre kevésbé gyorsan növekszik a körülfordulási idő.

### **3 Összefoglalás**

A cikkben bemutatunk egy új elosztott rendszerű alkalmazás szintű forgalomgenerátor rendszert, mely kiválóan alkalmas alkalmazások forgalmi modelljeinek gyors és hatékony fejlesztésére magas szintű nyelven. Természetesen ez csak egy hasznos funkciója a többi között. A fő célunk a egy pontos modellekkel rendelkező sokoldalúan használható forgalomgeneráló és hálózattervező eszköz kifejlesztése. A jelenlegi stádiumban egyelőre csak egyszerűbb méréseket tudunk végezni, mégis már most értékes eredményeket kaptunk, melyekből néhányat prezentáltunk.

#### 4 Irodalomjegyzék

- [1] S. Kesav, "REAL 5.0 Overview", Cornell University, 1997 , <http://www.cs.cornell.edu/skeshav/real/overview.html>
- [2] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, D. Zappala. "Improving Simulation for Network Research" Technical Report 99-702, University of Southern California, March, 1999 , <http://www.isi.edu/~johnh/PAPERS/Bajaj99a.html>
- [3] B. Viken, "Passive Monitoring of Internet Traffic at Supercomputing'98", Eunice'99, August 1999, [http://moat.nlanr.net/Papers/eunice\\_color.pdf](http://moat.nlanr.net/Papers/eunice_color.pdf)
- [4] M. Mathis, J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool" presented at [INET'96](#) in Montreal, Quebec 1996, <http://www.psc.edu/~mathis/ippm/>
- [5] A. Adams, J. Mahdavi, M. Mathis and V. Paxson, "Creating a Scalable Architecture for Internet Measurement", Pittsburgh Supercomputing Center and Berkeley National Laboratory, [http://www.psc.edu/~mahdavi/nimi\\_paper/NIMI.html](http://www.psc.edu/~mahdavi/nimi_paper/NIMI.html)
- [6] W. Matthews, L. Cotrell, "The PingER Project: Active Internet Performance Monitoring for the HENP Community", IEEE Communications Magazine on Network Traffic Measurements and Experiments, May, 2000, <http://www-iepm.slac.stanford.edu/paperwork/ieee/ieee.pdf>
- [7] V. Jacobson, "PATHCHAR, A tool to infer Characteristics of Internet Paths", Lawrence Berkeley National Laboratory, April, 1997, <ftp://ftp.ee.lbl.gov/pathchar/msri-talk.ps.gz>.
- [8] K. Lai and M. Baker, "Measuring Bandwidth", *Proceedings of IEEE INFOCOM '99*, March 1999, <http://mosquiconet.stanford.edu/~laik/projects/nettimer/index.html>
- [9] k claffy, Sean McCreary, UCSD, Internet measurement and data analysis: passive and active measurement, Statistical Computing and Graphics, Volume 10 No. 1, <http://www.caida.org/outreach/papers/Nae/4hansen.html>
- [10] M. J. Muuss, T. Slattery, (Mentor Technologies), "Test TCP: TCP Troughput Testing Tool", <http://www.mentortech.com/learn/tools/tools.shtml>
- [11] Hewlett-Packard Company, "Netperf: A Network Performance Benchmark, Revision 2.0", Information Networks Division, Hewlett-Packard Company, 1995, <http://www.netperf.org/netperf/training/Netperf.html>
- [12] C. Alexander, SPEC Standard Performance Evaluation Corporation "SPECweb96 Release 1.0", White Papers. <http://www.specbench.org>

- [13] J. Lewitt, *"Measuring Web-Server Capacity"*, INFORMATIONWEEK, January 1997, <http://www.specbench.org/osg/web96/infoweb/>
- [14] Crovella, M., Barford, P. *"Generating Representative Web Workloads for Network and Server Performance Evaluation"*, Tech Report BU-CS-97-006, , Boston University, December 1997, <http://www.cs.bu.edu/faculty/crovella/paper-archive/sigm98-surge.ps>
- [15] B. A. Mah, P. Sholander, L. Martinez, L. Tolendino *"IPB : An Internet Protocol Benchmark Using Simulated Traffic"*, Sandia National Laboratories, Albuquerque, NM, USA, <http://dlib.computer.org/conferen/mascots/8566/pdf/85660077.pdf>
- [16] Murayama, Y., Yamaguchi, S. *"DBS: a powerful tool for TCP performance evaluations"*, Performance and Control of Network Systems, Proceedings of SPIE, Volume 3231, Nara Institute, November 1997, <http://hayate.aist-nara.ac.jp/member/yukio-m/dbs/>
- [17] Lee, B. O., Frost, V. S., Jonkman, R. *"Netspec 3.0 Source Models for telnet, ftp, voice, video and WWW traffic"*, Tech Report ITTC-TR-10980-19, The University of Kansas, January 1997, <http://www.tisl.ukans.edu/Projects /AAI/products/netspec>