

# HTTP TRAFFIC SIMULATION AND EVALUATION FOR MULTIPLE USERS IN AN INTRANET NETWORK

Elias Aravantinos

Petros Ganos

Research Academic Computer Technology Institute

Riga Feraiou 61

GR- 26221 Patras, Greece

E-mail: eliasara@cti.gr

E-mail: ganos@cti.gr

Christos Bouras

Panagiotis Kokkinos

Computer Engineering and Informatics Department

University of Patras

GR-26500, Patras, Greece

E-mail: bouras@cti.gr

E-mail: kokkinop@ceid.upatras.gr

## KEYWORDS

IP Based Networks and Services, HTTP, Quality of Service, RMI, Web Server and Simulation.

## ABSTRACT

In this paper we describe the architecture and design of a network-stressing tool developed to simulate and evaluate the HTTP requests and responses of hundreds of users. There are two major issues that have to be considered when designing and implementing such a simulation tool, the reliability evaluation and the user "friendly" environment. In our application we use a mechanism exploiting Intranet technology to stress the tested network and confirm the performance, proper operation and delay issues. We have also implemented an easy interaction with the user considering the creation of the scenarios and the presentation of the simulation results, pointing out the problems and the network's status. Finally, we examined through a number of experiments using different simulation scenarios the behavior of the Intranet by load balancing the simulated users.

## INTRODUCTION

The purpose of this paper is to evaluate network performance during the design of an intranet network, containing hundreds of terminals, for enterprise buildings, airplanes, ships etc. There are certain parameters like bandwidth usage, and delay defining network performance. The Web traffic is considered as the main component of Internet backbone traffic aiming to assess the Web behaviour of the network. In this paper we aim to measure the load of HTTP traffic using a new HTTP simulator called NHS (Network HTTP Simulator). The NHS was the software developed to measure network performance under different conditions.

There are several papers that studied the wide range of network problems, reliability and HTTP simulation (Davison 2001), (Floyd 1999), (Heidemann et al. 1997), (Judge et al. 1998), (L. Breslau et al. 2000), (Rosu et al. 2000), (Feldmann et al. 1999), (Krishnamurthy and Rexford 1998), (Badrinath et al. 2000). The main papers related to

our work are described in the following paragraphs. The first paper (Davison 2001) describes the NCS HTTP simulator. This simulator tests HTTP 1.0/1.1 protocols in Ethernet, Fast Ethernet and modem environment. The NCS estimates client-side latencies and bandwidth usages. It also provides credibility; comparing simulation results to real world results and contains optional pre-fetching techniques. PROXIM is a caching and network effects simulator developed by researchers at AT&T Labs (Feldmann et al. 1999). It simulates a proxy cache using the HTTP trace as input. PROXIM can be used to simulate the three different scenarios using a proxy, without a proxy, where the bandwidth to the clients is the bottleneck, and without a proxy, where the bandwidth on the network connecting the clients to the Internet is the bottleneck. To assess the performance impact of proxy caching in a given environment, the simulation results are compared with-proxy to the without-proxy. In addition, the without-proxy simulation results are used to compare the simulation results against the original measured numbers. Network Simulator (NS) (Floyd 1999) is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The NS simulator is likely the best-known networking simulator, but is not typically used for caching performance measurements, possibly because of slow simulation speeds. It uses detailed models of networking protocols to calculate performance metrics. This simulator has been validated by widespread user acceptance and extensive verification tests. NS is not a polished and finished product, but is the result of an on-going effort of research and development. In particular, significant bugs in the software are still being discovered and corrected. NHS is described in this paper as a HTTP simulator, measuring HTTP traffic and presenting network performance. The Web Data in the simulated network are disseminated using the HTTP protocol. The NHS is a model independent of network technology and device configuration, flexible, scalable providing 'smart' Graphical User Interface (GUI); the user configures the simulation parameters easily in different scenarios. The

terminals are informed automatically from a central point about the simulation parameters in a single process.

This paper is organized as follows: First, the simulation design of NHS is described. Next the phase of simulation and functionalities are presented in details. The next section describes the proposed simulation model, containing techniques and specifications. In the following part the simulation results are presented and discussed. Finally the concluding remarks are provided.

## SIMULATION DESIGN

A general architecture of the simulation design is presented in Figure 1. The NHS user is located in the NHS server, configuring easily the simulation parameters from this central point. The simulation design is based on Remote Method Invocation (RMI) client server architecture.

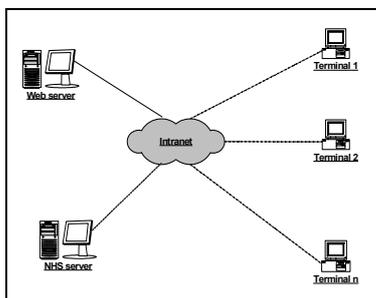


Figure 1: Network Architecture for HTTP Traffic

The NHS server hosts the server routine that communicates with the client routine hosted in the terminals. The NHS Server sends the parameters of simulation via RMI to the  $n$  terminals. A number of users are simulated in every terminal. During the simulation phase the terminals are connected to a Web Server via HTTP requesting Web pages. Each new Web page request creates a TCP connection. The Web Server responds to the Intranet, that consists of  $n$  terminals via HTTP. The NHS Server collects results of the simulation that are sent from the terminals and presents them to the user.

## NHS FUNCTIONALITIES

The operation of the NHS consists of three phases (Figure 2). The first is located at the NHS server; during this phase the user fills in various parameters and makes the selections involving the simulation scenarios. The second phase considers the NHS clients. Specifically during this phase HTTP traffic is created and several variables are measured. The third phase considers the NHS server and the presentation of simulation results to the user.

In the first phase the user defines various parameters and is guided to follow number of steps, like a wizard. In each step the user inserts values into the simulation parameters and configures the scenario according to the selections.

The parameters and the selections are the following:

- The number of users simulated in all client terminals,
- The IP addresses of the client terminals
- The IP addresses can also be loaded from a text file in case of large networks. In each of these terminals 'runs' the client routine
- The Web Pages that will be used in the HTTP simulation. These Web pages can be loaded from a text file to make the step easier.

It is possible to configure NHS to repeat automatically an experiment in a specific IP address, with the same parameters, but each time with fewer users simulated. The user can choose to conduct an experiment with the same parameters in all IP addresses.

Then the user fills in the values of the parameters that define the simulation scenario like IP address, the number of simulated users, the number of HTTP requests that each simulated terminal will make, the waiting time, which is the time between the arrival of response and the next request and the timeout. This is a time limit referring to the time spent trying to get the hosts IP-address, establishing a connection with the server and also setting the timeout on the socket while reading the response. If this time is expired an exception is sent.

In the second phase, the NHS server routine sends to each of the IP address, specified by the user a message. This message contains some of the parameters that were pre-defined and are necessary for the simulation process (initialization). These parameters, which are the input for the client routine, are already mentioned like timeout, waiting time etc.

After a client routine receives this packet (message) creates and starts a number of threads. A thread is a process that runs concurrently and independently from the routine that created it. Each thread corresponds to a passenger simulated in the specific IP address. Each of these threads makes a number of HTTP requests to the Web Server, requesting several Web pages. One very important aspect of the simulation is the specific process that a thread requests a Web page and gets the corresponding response. The NHS uses the HTTP1.1 protocol to communicate with the Web server. The NHS takes advantage of some specifications of HTTP1.1 protocol, such as the persistent connection, in order to send the HTTP requests. The NHS uses the persistent connection to open only one TCP connection per Web page, requesting through the same connection not only the html text, but also the images of the requested Web page. The advantage is to avoid opening a new TCP connection for each new HTTP request. During this whole process of sending HTTP requests and receiving the responses each thread stores some simulated data. Specifically the data that each thread stores are the following:

- The response code of each response, as defined in the HTTP1.1 protocol
- The request response time (page latency), for each request-response, includes the html page with the images

Figure 2: NHS Functionalities

- The first to last byte time for each response, which is the duration between the arrival of the first and the last byte of the response
- The number and the kind of errors, such as timeout expiration
- The number of bytes that each thread received
- The html pages that each thread received.

When all the threads complete the requests and receive the responses, the client routine that created these threads is finished. The next step is to send to the NHS server routine the results of all simulated threads. In the third phase the NHS server routine collects the results from all the client routines of the various terminals. Then the routine processes them and presents the final results to the user through the GUI. These results include:

- Log table for all HTTP requests. This table contains the response code, the first to last byte time and the request response time (page latency) of each request.
- General results table. This table contains average, minimum, maximum values of the request response time (page latency) referring to each IP address. Also contains total values of errors, pages and bytes.
- The general results table contains the bandwidth used by each IP address. The bandwidth is calculated by dividing the total number of bytes received in an IP by the sum of the first to last byte time of all requests hosted by this IP address.
- Initial parameters table. This table contains the parameters that the user defined in the first phase.
- Graphs, performing any errors, bandwidth usage (bytes/sec), request response time and request response time (page latency) for each Web page. The results are saved in a file (html format), creating an archive of previous experiments.

## IMPLEMENTATION ISSUES

NHS was implemented in Java. The NHS software consists of a number of classes, which are separated in to two different groups. The first group is the NHS server software, which runs in the NHS server, and its main task is the interaction with the user. The second group is the software that runs in each client terminal and its main task is to make the HTTP requests to the Web Server, to receive the responses, to keep a log of its operation and to measure various parameters involving simulation.

## EXPERIMENTS

The NHS was tested in the network topology shown in Figure3. This is also the worst and most complicated case of our experiments described in details in this section. All the tested terminals are in different segments to increase the complexity of NHS experiments.

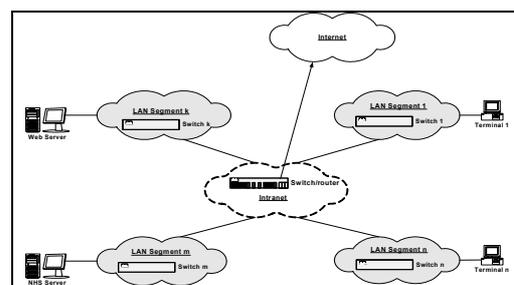


Figure 3: Network Topology

The tested network was an Intranet, supported by a Fast Ethernet network with star topology. The Fast Ethernet network cable is Unshielded Twisted Pair category 5 (UTP cat 5). The tested network consists of a certain number of terminals, servers and Local Area Network (LAN) switches. Each LAN segment is located in a different physical place in the Intranet. The Web Server responds to the client requests when download the Web pages. The NHS Server hosts the Simulator server routine that controls and communicates with the client-terminals. The terminals are single PCs. The specifications of the terminals and servers used in simulation tests are described in details in section 4 of this paper. The LAN switches are Fast Ethernet network devices that provide each sender/receiver pair with full 100 Mbps capacity. Each port on the switch gives full bandwidth to a single server or client station. Finally, there is a switch – router connected to the Internet to serve the Internet and Web based applications of the Intranet. The terminals and the servers are connected in different LAN segments. It was supposed that the tested terminals are located in different physical LANs instead of VLANs. All the terminals are connected to different switches instead of VLANs of the same switch. This view increases the complexity of NHS.

The terminals and the servers of the simulation had the following software specifications: The Web server supported Windows 2000 Server operating system and IIS 5.0, the NHS server contained Windows 2000 Server, IIS 5.0, JRE 1.3, NHS Server Routine and the terminals Windows 2000, JRE 1.3 package and the NHS Client Routine.

In the simulation phase several Web pages were used with different size (Kb), contained different number of images. The Web pages used, were 4 with total size between 38 and 140 KB, containing several number of pictures for example the first web page contains 25 pictures with total web page size 140 KB, the second 5 pictures with total size 38 KB, the third 16 pictures with 120 KB total size and the fourth 30 pictures with 121 KB size.

## SIMULATION RESULTS

The results presented below are separated in two categories. The results that were measured in experiment and presented in NHS GUI and the results and values which were collected from NHS and presented in charts.

Firstly the results, as measured in experiment and presented in NHS GUI:

This experiment was conducted by setting the following values to the simulation parameters:

- Terminals: 4
- Used Users simulated per terminal: 70
- HTTP Requests Per simulated user: 3
- Web Pages used: 3
- Waiting Time: 1.0 sec,
- Timeout: 10 sec

- HTTP Requests Per simulated user: 3,
- Web Pages used: 3,
- Waiting Time: 1.0 sec,
- Timeout: 10 sec

The Table 1 contains samples of the simulation results, which are presented in the following charts, as exported after the simulation in the log file of NHS. The RRT symbolizes the Request Response Time and refers to the page latency, which is the time between HTTP request and response:

Table 1: General Results Table

Bytes	Bandwidth	Max RRT	Min RRT	Avg RRT
22632680	90587.889	4025	81	1351
22632680	98037.662	3625	50	1302
22632680	107639.012	4026	60	1152
22632680	107507.600	3104	50	1147

The chart (Figure 4) performs the simulation of 280 users in four terminals - 70 users per terminal. The graph represents the average RRT (Request Response Time (millisecond) for each tested Web page in relation with each IP address (terminal). In this experiment 3 different Web pages were tested (Simulation model Section).

The RRT is increased as the size of the web page is also increased. However there is a deviation between the RRT of the first and the third Web page, although their size (Kilobytes) is close. In addition the corresponding page latencies of the second and the third Web page are close, although the size of the two web pages is different. This is due to the fact that in both Web pages the size of the html page (text) is extremely small (10 and 7 Kilobytes), in contrast to the first that is 75Kbs. The first Web page contains more images, needing more HTTP requests and increasing RRT. The RRT difference between almost the same size Web pages is due to the size of the html page and to the number of images of each Web page.

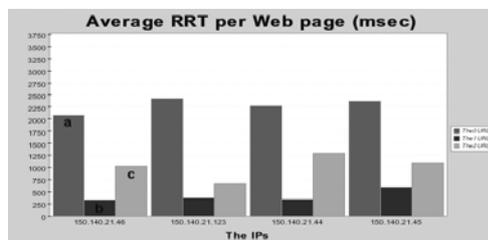


Figure 4: Results containing 4 terminals and 3 Web pages

Then we examined the results collected from the NHS processes, after a certain number of experiments: These experiments were conducted by setting the following values in the NHS parameters:

- Terminals: 4,
- Users simulated per terminal: 50/60/70/80/90,

The following charts (Figure 5) represent the bandwidth (bytes/sec) and the RRT in comparison to the number of users simulated. The several types of lines refer to the different IP addresses. In this chart are presented the measured results of 5 experiments - 50, 60, 70, 80 and 90 simulated users. We repeated the experiments, using the same parameters twice, in order to test the results.

The results lead to the conclusion, that the bandwidth of each terminal is decreased as the number of users simulated is increased. Certainly, this is a reasonable conclusion increasing the reliability of the NHS and the simulation process. From these charts we can conclude that the RRT is increased as the number of simulated users, in each IP, is also increased. It was also noted that the results, as measured, are very close.

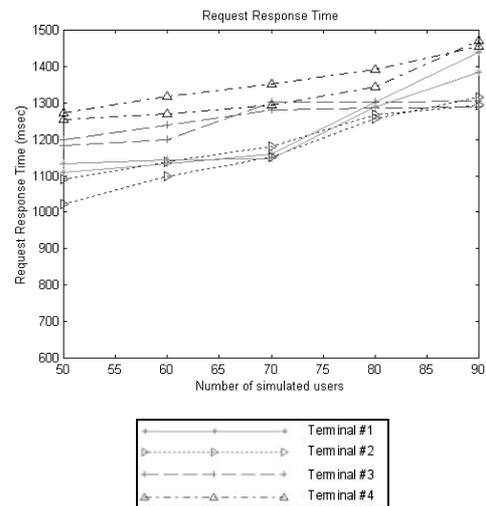


Figure 5: RRT Comparison of the Tests

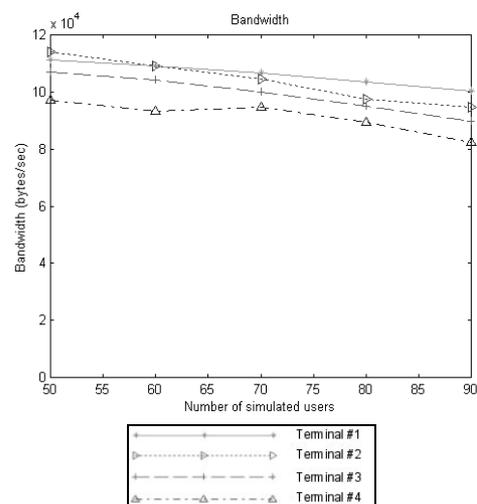


Figure 6: Bandwidth Comparison of the Tests

Finally we must remark that the previous experiments represent only some samples of the executed experiments.

Also the results of these experiments depend from the way that NHS functions. NHS operates like a browser although there are differences; in the same way there are differences between known browsers like Explorer and Netscape.

## **SIMULATION CONCLUSIONS AND FUTURE WORK**

NHS is a HTTP simulation tool, which can evaluate the reliability and verify the normal operation and performance of a LAN (intranet). The installation phase and operation of NHS are very easy. The components (servers and terminals) of the simulation process do not need to support any special hardware or software specifications.

The NHS user has a complete control during the simulation process, taking advantage of a very user-friendly GUI. So it is possible to set the values of several simulation parameters, creating different scenarios, to receive the results and assess the simulation process.

NHS is independent of the network topology where is executed. NHS can operate independently of the network topology and technology (ATM, Gigabit Ethernet) and the active network hardware and devices. The results of the simulation from all terminals are presented in a friendly and easy way in a central point, which is the NHS server (GUI). In the future it is planned to conduct more experiments, using some hundreds of terminals and simulating even more users in each terminal. Also it is planned to test NHS in different intranets, various network topologies (bus, ring) and technologies (ATM, Gigabit Ethernet). In addition the purpose is to reduce even more the load of memory and CPU usage that the NHS creates in each terminal. Furthermore we will try different request distributions including Zipf-like distributions (Li 92). Finally one of the targets is to improve the GUI, importing new characteristics and new parameters creating even more complicated scenarios.

## **REFERENCES**

- Badrinath B. R. and P. Sudame. Gathercast, 2000: The design and implementation of a programmable aggregation mechanism for the Internet. In Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN), Oct. 2000.
- Breslau L., D. Estron, K. Fall, S. Floyd, J. Heidemann, Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, 2000. Advances in network simulation. *IEEE Computer*, 33(5): pp 59–67, May 2000.
- Davison Brian D., 2001: HTTP Simulator validation using real measurements: A case study Published in Proceedings of Ninth International Symposium on Modeling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS '01), Cincinatti, August 2001.
- Feldmann A., R. Caceres, F. Douglis, G. Glass, and M. Rabinovich, 1999: Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments. In Proceedings of IEEE INFOCOM. pp. 106–116, New York, Mar. 1999.
- Floyd S., 1999: Validation Experiences with the NS Simulator. In Proceedings of the DARPA/NIST Network Simulation Validation Workshop, Fairfax, VA, May 1999.
- Heidemann J., K. Obraczka, and J. Touch, 1997. Modeling the performance of HTTP over several transport protocols. *IEEE/ACM Transactions on Networking*, 5(5): pp 616–630.
- Judge J., H. Beadle, and J. Chicharo, 1998. Sampling HTTP Response Packets for Prediction of Web Traffic Volume Statistics.
- Krishnamurthy B. and J. Rexford, 1998. Software Issues in Characterizing Web Server Logs. In World Wide Web Consortium Workshop on Web Characterization, Cambridge, MA, Nov.1998. Position paper.
- Li W., 1992: Random texts exhibit Zipf's-law-like word frequency distribution. *IEEE Transactions on Information Theory*, 38:1842–1845.
- D. Rosu, A. Iyengar, and D. Dias, 2000: Hint-based Acceleration of Web Proxy Caches. In Proceedings of the 19th IEEE International Performance, Computing, and Communications Conference (IPCCC 2000), Phoenix, AZ, Feb. 2000.