ISSN:0975-9646

Sandhyarani S et al / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2011, 2620-2626.

NETWORK DELAY: NETWORK ANALYZER AND OPNET SIMULATION TOOL

SANDHYARANI.S CHANDRASEKHAR.G

Department of Computer Science and Engineering
Joginpally B.R Engineering college
Moinabad — 500075

Abstract— **This paper presents a complete network analyzer development for network delay in campus environment. The purpose of this study is to define the accuracy of network analyzer development with independent data, real network and OPNET simulation tool. The network delay will measure based on transmission delay and propagation delay. This network analyzer software will test on delay generated by the several services. The reliability of this network analyzer will test with email, text messaging and instant messaging over web service. The results show that network analyzer software has accuracy and same trend delay with independent data, real network and OPNET simulation tool. Finally, this software is able to measure the network delay during preparation, proposal and planning phases.**

Keywords: *Accuracy, Traffic, Network Analyzer, OPNET*

1. INTRODUCTION

This study focuses on the accuracy of network delay using heterogeneous services. This stud does not intend to perform a comprehensive test the functionality of all simulator and analyzer features. OPNET has originally been developed for network simulation and it is fully usable as a robust and reliability simulation tool with higher investment. This network analyzer development process has discussed detail in [1], [2], [3].Service performance requirement consists of delay, capacity (bandwidth) and reliability [4]. Reliability is a measure of the network/system ability to provide deterministic and accurate delivery of information [5], [6]. Delay is a time difference in transmitting a single unit of information (bit, byte, cell, frame, and packet) from source to destination.

2. DEFINITION OF NETWORK ANALYZER

A network analyzer also called a "packet analyzer," "traffic analyzer" and "protocol analyzer," [7]. Network analyzers functionality such as [8]: i) provide detailed statistics for current and recent activity on the network; ii) detect unusual levels of network traffic; iii) detect unusual packet characteristics; iv) identify packet sources or destinations; v) configure alarms for defined threats; vi) and vii) monitor bandwidth utilization.

Network Management: Network management consists of a variety of tasks, for example, monitoring, configuration, troubleshooting and planning that are performed by users and network administrators [9]. Network element is a component of the network that can be managed. This includes hosts, routers, switches, hubs and server that can be measured. Examples of end-toend characteristics for network elements and network traffic are capacity (bandwidth), availability, delay, jitter, throughput, network utilization and error rates [10], [11].

2.1 Introduction to Network Analyzer

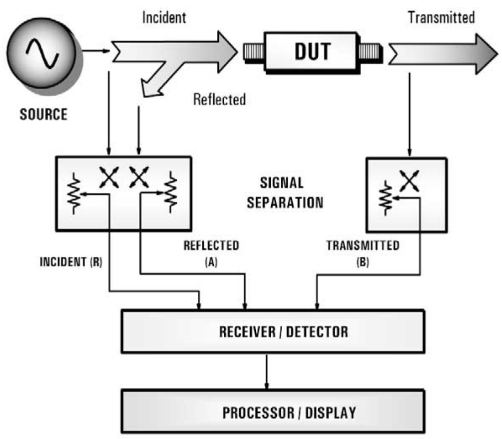
A network analyzer (also called a protocol analyzer or packet analyzer) is a combination of hardware and programming, or in some cases a stand-alone hardware device, that can be installed in a computer or network to enhance protection against malicious activity. Network analyzers can supplement firewalls, antivirus.

Network analyzers can:

- Provide detailed statistics for current and recent activity on the network.
- Test anti-malware programs and pin point potential vulnerabilities
- Detect unusual levels of network traffic.
- Detect unusual packet characteristics.
- Identify packet sources or destinations.
- Configure alarms for defined threats.
- Search for specific data strings in packets.
- Monitor bandwidth utilization as a function of time.
- Create application-specific plug-ins.
- Display all statistics on a user-friendly control panel

- Network analyzers are not intended to replace firewalls, anti-virus programs, or spyware detection programs. However, the use of a network analyzer in addition to other countermeasures can minimize the probability that an attack will occur, and can facilitate rapid response in the event an attack begins.

2.2 Generalized Network Analyzer Block Diagram



Here is a generalized block diagram of a network analyzer, showing the major signal-processing sections. In order to measure the incident, reflected and transmitted signal, four sections are required:

- Source for stimulus
- Signal-separation devices
- Receivers that down convert and detect the signals
- Processor/display for calculating
- and reviewing the results

3. METHODOLOGY

Figure 1 shows network life cycle approach for technologies and services implementation in the future [12]. Network life cycle approach consists of six phases such as prepare, plan, design, implement, operate and optimize. This network analyzer development concentrates more on preparation, planning and proposal areas. Network analyzer development is based on mathematical model. We use queuing theory M/M/1 to build this software [13], [14]. The reliability test will only concentrate on network delay.

The independent data output is generated based on number of distance input, number of nodes input, size of bandwidth input and size of services input. These inputs will use in OPNET, real network and network analyzer software (refer to Figure 3).

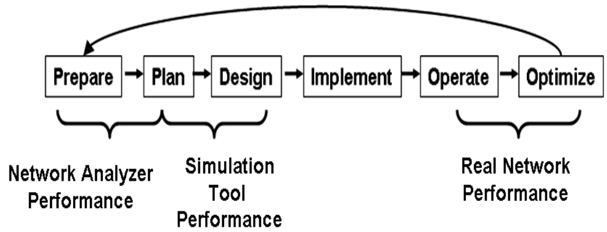


Figure 1: Comparison of Network Analyzer, OPNET and Real Network

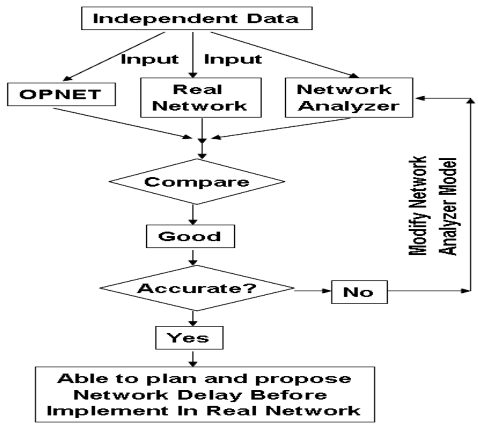


Figure 2 shows network delay reliability test.

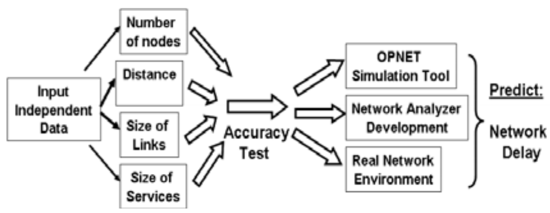


Figure 3: Input Parameters for Network Analyzer, OPNET and Real Network

4. NETWORK SIMULATION

Simulation Modeling is becoming an increasingly popular method for network performance analysis. Generally, there are two forms of network simulation: analytical modeling and computer simulation. The first is by mathematical analysis that characterizes a network as a set of equations. The main disadvantage is it's over simplistic view of the network and inability to simulate the dynamic nature of a network. Thus, the study of a complex system always requires a discrete event simulation package, which can compute the time that would be associated with real events in a real-life situation. Software simulator is a valuable tool especially for today's network with complex architectures and topologies. Designers can test their new ideas and carry out performance related studies, therefore freed from the burden of the "trial and error" hardware implementations.

REAL: REAL is a simulator for studying the dynamic behavior of flow and congestion control schemes in packet switch data networks. Network topology, protocols, data and control parameters are represented by *Scenario,* which are described using Net Language, a simple ASCII representation of the network. About 30 modules are provided which can exactly emulate the actions of several well known flow control protocols (S. Keshav 1997).

INSANE: INSANE is a network simulator designed to test various IP-over-ATM algorithms with realistic traffic loads derived from empirical traffic measurements. It's ATM protocol stack provides real-time guarantees to ATM virtual circuits by using Rate Controlled Static Priority (RCSP) queueing. A protocol similar to the Real-Time Channel Administration Protocol (RCAP) is implemented for ATM signalling. A Tk-based graphical simulation monitor can provide an easy way to check the progress of multiple running simulation processes (INSANE Homepage).

NetSim: NetSim is intended to offer a very detailed simulation of Ethernet, including realistic modeling of signal propagation, the effect of the relative positions of stations on events on the network, the collision detection and handling process and the transmission deferral mechanism. But it cannot be extended to address modern networks (Lewis, Barnett 1993).

Maisie: Maisie is a C-based language for hierarchical simulation (L. Bagrodia 1991), or more specifically, a language for parallel discrete event simulation. A logical process is used to model one or more physical processes; the events in the physical system are modeled by message exchanges among the corresponding logical processes in the model. User can also migrate into recent extension: Parsec and MOOSE (an object-orient extension) (Rajive Bagrodia 1995).

5. OPNET SIMULATOR

OPNET (Optimized Network Engineering Tool) provides a comprehensive development environment for the specification, simulation and performance analysis of communication networks. A large range of communication systems from a single LAN to global satellite networks can be supported. Discrete event simulations are used as the means of analyzing system performance and their behavior. The key features of OPNET are summarized here as:

- **Modeling and Simulation Cycle** OPNET provides powerful tools to assist user to go through three out of the five phases in a design circle(i.e. the building of models, the execution of a simulation and the analysis of the output data), see Figure 1.

- **Hierarchical Modeling** OPNET employs a hierarchical structure to modeling. Each level of the hierarch describes different aspects of the complete model being simulated.
- **Specialized in communication networks** Detailed library models provide support for existing protocols and allow researchers and developers to either modify these existing models or develop new models of their own.

- **Automatic simulation generation** OPNET models can be compiled into executable code. An executable discrete-event simulation can be debugged or simply executed, resulting in output data.

5.1 Hierarchical Modeling

OPNET provides four tools called editors to develop a representation of a system being modeled. These editors, the Network, Node, Process and Parameter Editors, are organized in a hierarchical fashion, which supports the concept of model level reuse. Models developed at one layer can be used by another model at a higher layer. Figure 2 portrays this hierarchical organization. The following sections introduce each of the modeling domains. The Parameter Editor is always seen as a utility editor, and not considered a modeling domain.

5.1.1 Network Model

Network Editor is used to specify the physical topology of a communications network, which define the position and interconnection of communicating entities, i.e., *node* and *link*. The specific capabilities of each node are realized in the underlying *model*. A set of parameters or characteristics is attached with each model that can be set to customize the node's behavior. A node can either be fixed, mobile or satellite. Simplex (unidirectional) or duplex (bi-directional) point-to-point links connects pairs of nodes. A *bus link* provides a broadcast medium for an arbitrary number of attached devices. Mobile communication is supported by *radio links*. Links can also be customized to simulate the actual communication channels.

The complexity of a network model would be unmanageable where numerous networks were being modeled as part of a single system. This complexity is eliminated by an abstraction known as a *subnetwork*. A subnetwork may contain many subnetworks, at the lowest level, a sub network is composed only of nodes and links Communications links facilitate communication between subnetworks.

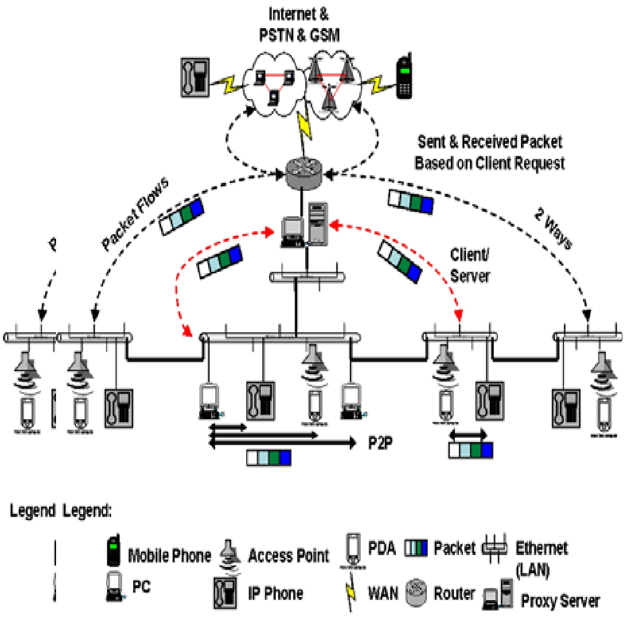


Figure 4: Real Heterogeneous Network Environment at Main and Branch Campus

5.1.2 Node Model

Communication devices created and interconnected at the network level need to be specified in the node domain using the Node Editor. Node models are expressed as interconnected *modules*. These modules can be grouped into two distinct categories. The first set is modules that have predefined characteristics and a set of built-in parameters. Examples are packet generators, point-to-point transmitters and radio receivers. The second group contains highly programmable modules. These modules referred to as *processors* and *queues*, rely on process model specifications.

Each node is described by a block structured data flow diagram. Each programmable block in a Node Model has its functionality defined by a Process Model. Modules are interconnected by either *packet streams* or *statistic wires*. Packets are transferred between modules using packet streams. Statistic wires could be used to convey numeric signals.

5.1.3 Process Model

Process models, created using the process editor, are used to describe the logic flow and behavior of processor and queue modules. Communication between process is supported by *interrupts*. Process models are expressed in a language called Proto-C, which consists of state transition diagrams (STDs), a library of kernel procedures, and the standard C programming language. The OPNET Process Editor uses a powerful state-transition diagram approach to support specification of any type of protocol, resource, application, algorithm, or queueing policy. States and transitions graphically define the progression of a process in response to events. Within each state, general logic can be specified using a library of predefined functions and even the full flexibility of the C language. Process may create new processes (*child process*) to perform sub-tasks and thus is called the *parent process*

6. SIMULATION MODEL DEVELOPMENT FOR REMOTE DATA TRANSFER

Many different types of modeling and simulation applications are used in various disciplines such as acquisition, analysis, education, entertainment, research and training [14]. In the Figure 4.1, theoretical model is based on a random distribution of service duration. Simulation model is divided as follows: i) to study physical model of real heterogeneous network environment; ii) transform physical model of real heterogeneous network environment into logical model; and iii) develop and implement the heterogeneous simulation model.

6.1 PHYSICAL MODEL OF REAL LAN HETEROGENEOUS ENVIRONMENT

Figure 4 shows the network heterogeneous environment in real world. The physical model of real LAN heterogeneous environment is based on multi-services and multi-technologies has implemented at University of Kuala Lumpur. Then we need to transform from heterogeneous environment in realworld into logical model. The logical model is the phase where mathematical techniques are used to stimulate heterogeneous environment.

7. OPNET ANALYSIS TOOL

Simulations can be used to generate a number of different forms of output, as described above. These forms include several types of numerical data, animation, and detailed traces provided by the OPNET debugger. In addition, because OPNET simulations support open interfaces to the C language, and the host computer's operating system, simulation developers may generate proprietary forms of output ranging from messages printed in the console window, to generation of ASCII or binary files, and even live interactions with other programs. However, the most commonly used forms of output data are those that are directly supported by Simulation Kernel interfaces for collection, and by existing tools for viewing and analysis. Both animation data and numerical statistics fall into this category. Animation data is generated either by using automatic animation probes or by developing custom animations with the KP's of the Simulation Ker-nel's Anim package; the m3_vuanim utility is then used to view the animations. Sim-ilarly, statistic data is generated by setting statistic probes, and/or by the KP's of the Kernel's Stat package; OPNET's Analysis Tool can then be used to view and manip-ulate the statistical data.

The service provided by the Analysis Tool is to display information in the form of graphs. Graphs are presented within rectangular areas called analysis panels. A number of different operations can be used to create analysis panels, all of which have as their basic purpose to display a new set of data, or to transform an existing one. An analysis panel consists of a plotting area, with two numbered axes, generally referred to as the abscissa axis (horizontal), and the ordinate axis (vertical). The plotting area can contain one or more graphs describing relationships between variables mapped to the two axes. For example, the graph in the panel below shows how the size of a queue varies as a function of time.

7 ACCURACY OF SIMULATION MODEL WITH REAL LAN EXPERIMENTAL

In this section, we verify the little law and queuing theories for heterogeneous simulation model experiments. The experiments are composed of WAN and LAN experiment to real LAN environment. Several tests were performed to evaluate the tuned parameters and the values that better 'mimic' the characteristics of the real networks. Real experiment is based on real network and need to consider as follows: i) network bandwidth is limited and is not enough for all application and users at the same time; ii) delay due to the network overloads; and iii) packet losses.

7.1 EXPERIMENTAL OF REAL NETWORK SETUP

We used a network management application (Colasoft Capsa) to capture traffic between two networks link in real LAN experiments. Figure 5 shows the experimental setup of real network used in our tests. Fluke Optiview device can be configured to insert size of packet services and number of clients to generate traffic into the network interface port By using varying number of clients and size of packet services, we are able to simulate network traffic.

8. OPNET FILTER TOOL

OPNET's Analysis Tool allows the user to extract data from simulation output files and to display this data in various forms, as described in Chapter Data n of the OPNET Modeling Manual. The Analysis Tool also supports several mechanisms for numerically processing the data and generating new data sets that can also be plotted. These include computing probability density functions and cumulative distribution functions, as well as generating histograms. The data presented in the Analysis Tool may also be operated on by numeric filters. These are constructed from a pre-defined set of filter elements in the Filter Editor.

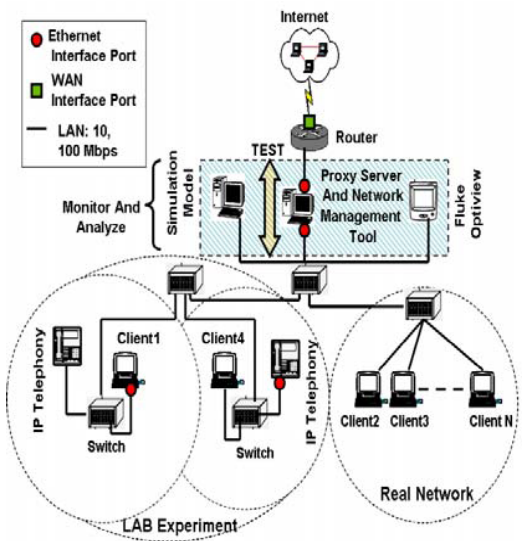


Figure 5: Experimental Laboratory for Real LAN Environment Setup

Filter models are represented as block diagrams consisting of interconnected filter elements. Filter elements may be either built-in numeric processing elements, or references to other filter models. Thus, filter models are hierarchical, in that they may be composed of other filter models. However, all filter models must be composed at the lowest level of pre-defined filters discussed in Chapter Data n of the OPNET Modeling Manual. Filters operate on vectors. Vectors are discrete and ordered sets of numeric data which consist of entries, as discussed in Chapter Data n of the OPNET Modeling Manual. Each entry consists of an abscissa and an ordinate value. These are double-precision floating point numbers. A filter model may operate on one or more vectors and combine them to form its output, which must consist of just one vector. The vectors that are fed into the filter are called input vectors; the result of the filter's processing is called the filter's output vector.

9. SIMULATION EXAMPLE

First, we give out a queueing network example as shown in Figure6. There are three FIFO queues in tandem and each has several homogeneous sources. We can specify build in parameters for the source and queue, respectively. The source will generate constant length packets in a Poisson manner with a given rate. The buffer capacity and service rate for each queue can also be specified. By changing the buffer capacity and service rate, some simulation results are shown in Figure 7 to Figure 9. The end-to-end delay and loss ratio against different service rate values are given out in Figure 7 and Figure 8, respectively.

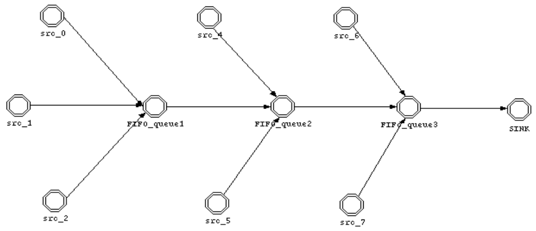


Figure 6: An example of queueing network

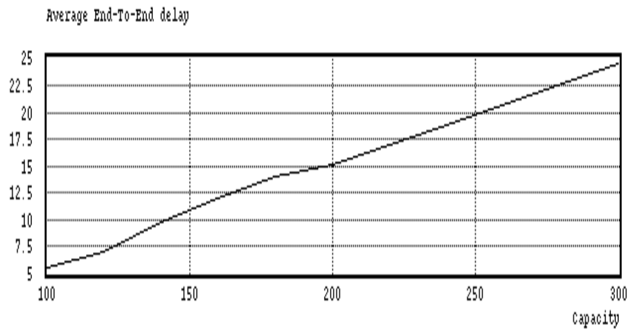


Figure 7: End-to-end delay vs. queue buffer capacity

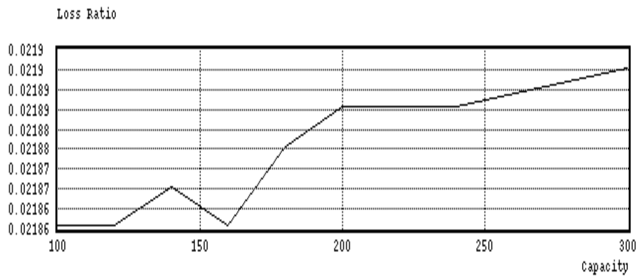


Figure 8: Loss ratio vs. queue buffer capaci

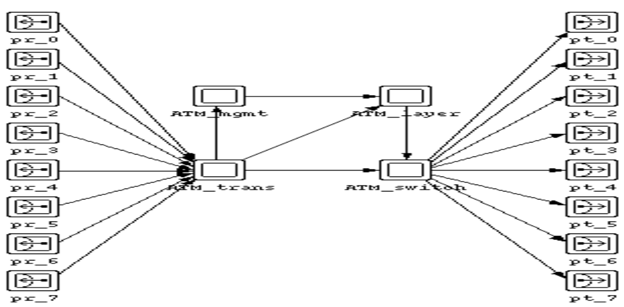


Figure 9: Node of Ethernet workstation and ATM Switch

10. CONCLUSION AND FUTURE WORK

In this article, we have shown how an analytical queuing network model can be used to understand the behaviors of heterogeneous environment over Wide Area Network, WAN (real network). The most apparent aspect is the delay due to the propagation and transmission time. Our simulation model, has demonstrated that it can measure accurately the performance of heterogeneous services and technologies to transfers data between two networks. Through WAN (real network) experiments, the simulation model is verified and validated for providing accurate performance information for various services. We believe the simulation modeling framework described in this study can be used to study other variations, tunings, and similar new ideas for various services and technologies. In network management, by monitoring and analyzing network delay we can monitor the performance of the network, thus to study whether network is normal, optimal or overloaded. Future work is to develop a simulation model to analyze bandwidth capacity requirement for various services and technologies in heterogeneous environment.

REFERENCES

1. M. Fleury, G. Flores Lucio, and M. J. Reed “Clarification of the ‘OPNET NS-2 Comparison’ Paper with. regards to *OPNET* Modeler”. http://privatewww.essex.ac.uk/~fleum/OPNET NS2_Comparison.pdf, accessed 9/9/2008.

2. Brown K, Christianson L, OPNET Lab Manual to Accompany Data and Computer Communications 7th ed. And Computer Networking with Internet Protocols and Technology 4th ed. by William Stallings, Pearson Prentice Hall, Upper Saddle River, NJ, 2005.

3. Xinjie Chang. 1999. “Network Simulations With OPNET”. Proceedings of the 1999 Winter Simulation Conference P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

4. James D. McCabe. “Practical Computer Network Analysis and Design”, pp. 15 –40, 1998, Morgan Kaufmann Publishers.

5. Michael R. Lyu. “Software Reliability Engineering: A Roadmap”, International Conference on Software Engineering, pp. 153-170, 2007.

6. Abdullah Konak. “Combining network reductions and simulation to estimate network reliability”, Proceedings of the 39th conference on Winter simulation, pp. 2301-2305, 2007.

7. Ulf Jonsson. “100BASE-X over SMF 5 Criteria, 2002”, (Online),

8. Victor A. Clincy & Nael Abu-Halaweh. 2005. “A Taxonomy of free Network Sniffers for teaching and research”, Source” Journal of Computing Sciences in Colleges Vol. 21(1), pp. 64 – 75.

9. J. P. Talledo. 2005. “Design and Implementation of an Ethernet Frame Analyzer for High Speed Networks”, Proceedings of the 15th International Conference on Electronics, Communications and Computers, Publisher IEEE Computer Society pp. 171 – 176.

10. Jairo A. Gutiérrez. “A connectionless approach to integrated network management”, International Journal of Network Management Vol. 8(4), pp. 219 – 226, 1998, John Wiley & Sons, Inc. New York, NY, USA

11. Faouzi Kamoun. “Toward best maintenance practices in communications network management”, International Journal of Network Management Vol. 15(5), pp. 321 – 334, 2005, John Wiley & Sons, Inc. New York, NY, USA

12. Cisco Networks. 2007. “Cisco’s Life Cycle Services Approach”, Cisco Networkers Conference 07, Januari 2007, Sunway Pyramid, Malaysia.

13. Mohd Nazri Ismail, Abdullah Mohd Zin, "Comparing the Accuracy of End-to-End Network Performance Measurement Testbed and Simulation Model for Data Transfers in Heterogeneous Environment," ams, pp. 124 131, IEEE, Second Asia International Conference on Modelling & Simulation (AMS), 2008.

14. Mohd Nazri Ismail and Abdullah Mohd Zin. “Development of Simulation Model in Heterogeneous Network Environment: Comparing the Accuracy of Simulation Model for Data Transfers Measurement over Wide Area Network”. Asian Network for Scientific Information. Information Technology Journal, pp. 2448, 2008. ISSN: 1812-5638 (Print). Pakistan.

15. Hans Lohninger. “Wireless Networking in the Developing World”, January 2007, [Online access], http://www.vias.org/wirelessnetw/wndw_05_07.html

16. David Grangier. “Transmission versus Propagation Delay“, Eurecom Institute, [Online access], **http://www.cse.ust.hk/**