Fourth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCET'2006)
"Breaking Frontiers and Barriers in Engineering: Education, Research and Practice"
21-23 June 2006, Mayagüez, Puerto Rico.

Modeling Web-Based Pediatric MRI Data Repository Site Using OPNET

Magno R. Guillen. PhD Candidate
Research Assistant CATE, Florida International University, Miami, Fl, USA, magno.guillen@fiu.edu

Malek Adjouadi, PhD
CATE Director, Florida International University, Miami, Fl, USA, adjouadi@fiu.edu

William D. Gaillard, M.D.
Department of Neurology and Pediatrics, Children's National Medical Center, The George Washington University, Washington, DC, GaillardW@ninds.nih.gov

Abstract

In the last years, many efforts have been conducted to study the spatio-temporal behavior of brain activation patterns while performing controlled tasks. However, since each institution has a limited subject population to conduct the study, the statistical analysis has been very limited. The creation of a multisite MRI data repository would provide substantial data to convey studies over a widely distributed demography.

This paper reports the development of a web based platform that contains MRI images and background information taken from subjects at participating hospitals here in The United States and overseas as part of this multisite consortium. The data is stored in a Network Attached Storage (NAS) system with Server Load Balancing (SLB) features. This site in its current design reveals a fast response time, scalability and reliability that together yield high performance in data access and availability.

A solution based on a Linux platform with Apache, My-SQL and PHP is proposed. Several SBL polices have been analyzed. To validate the performance of this new design, a simulation using the Optimized Network Engineering Tool (OPNET) has been performed, and the results obtained support the assertion that the design approach considered represents an effective and reliable method for the proposed multisite.

Keywords: Load Balancer, Data Base Server, Web Server, Dicom, Pacs

1. Introduction

For the medical research community, the availability of large data sets that are easily accessible for their research needs is fundamental not only in the added value in terms of better data mining prospects but also in the consequential diagnosis and prognosis processes that can result data. Everyday thousands of Magnetic Resonance Images (MRI) are generated, but most of this wealth of information is not available for researchers to exploit.[1] It is clear that the more information is known about the background of the subjects, together with the clinical profiles and the circumstances under which the tests were performed, the better it will be to identify and understand the varied aspects and specific characteristics of the test result data.

The creation of a centralized database that holds MRI images, and its aggregated information, is therefore most welcome by the community of researchers and healthcare providers. For this main reason, this paper reports the simulation result of a web based platform that contain MRI images and general medical background information taken from subjects at hospitals that are currently part of the consortium. This consortium is expected to grow and will extend to other hospitals wishing to join as the multisite matures.

A DIRECTOR server is set up to access image information or medical data in a balanced approach. The database for images is set up such that it can be distributed in several back end database servers with a centralized data repository. A NAGIOS server or dedicated hardware can also be used to do the redirection of queries (Nagios, 2005). Once fully developed, this computational infrastructure will support research studies in healthcare industry using existing shared information and computational resources. Moreover, this infrastructure would provide on demand computing for on-line and off-line real data processing, modeling and simulation to be used in clinical decision intelligence (CDI) applications. Since the data will be stored in a Network Attached Storage Area Network (NAS) system with Server Load Balancing (SLB) features, this site would most importantly have a fast response time, and will be scalable, reliable and with maximized throughput.

2. Related Work

In 1995, Chu and colleagues (Chu et al., 1995) presented the idea of a model to query formulation over clinical information systems including Hospital Information System (HIS), Radiology Information System (RIS) and Picture Archiving and Communication System (PACS) data, to look for spatial relationship between objects and temporal relationships between objects and events. This work resulted in a knowledge-based multimedia medical distributed database System.

In 1998, a unified approach to handle medical images was proposed by Lin and colleagues (Lin et al., 1998). This approach introduced the idea of using web based interface to access the patient medical records and support telemedicine.

In 2001, a web based telemedicine system was proposed that allowed the exchange of patient information records using a java platform. This system would serve not only as a remote tool to provide consultation between doctors and discuss case studies among several hospitals, but also as a patient oriented telemedicine system for the health care system. This system included "case submission", "medical image loading", "case diagnosis", "case presentation" and "case consultation" modules (Ahmed et al., 2001). The same year, Yu Lim proposed a cost effective PACS system with web-based interfaced implemented in Java and Common Gateway Interface (CGI) scripts. This project resulted in a web-based collaborative system for medical image analysis and diagnosis providing a Computer Supported Collaborative Work (CSCW) tool in medical area. (Lin et al., 2001)

In 2003, Munch and colleagues (Munch et al., 2003) proposed a web-based distribution of radiology images integrating medical images from the PCAS with medical information from EPR and making then

[1] This paper is in the framework of the **Multi-Site Pediatric Network for fMRI Mapping in Childhood Epilepsy** Proposed by William D. Gaillard MD, PI Children's National Medical Center

available for consultation in the intranet and the internet. The proposed system used the Java Advanced Imaging Application program Interface (JAIAPI) and it was developed in JAVA.

Through this prior work, noted progress has been made in this area of research, but there remains a lack of stronger evidence in optimizing the performance and configuration of the hardware and software platform and their eventual assimilation for a highly integrated design. In this paper a careful design and configuration analysis study is performed to respond to this need.

3. Scope of the work

In general, designing a website to provide access to an MRI database is a problem that involves not only the performance characteristics of the site itself but the hardware and software platform this site is built upon. Inherent to any good website design are the two cornerstones of scalability and reliability. Therefore, the selection of the appropriate hardware along with the related software developments will have to be carefully weighed.

There are at least 10 health care institutions willing to collaborate providing patient medical records, with an average of 30 patients per site per year, and an average of 125 images per study, each image size is 512 by 512 by 16 bits. This produces an estimate of 65.5 MB of data per subject study, having 5 studies per patient as average. This is expected to reach a data size of 9.8GB per institution or an estimated 98 GB of data per year from the 10 institutions.

Perhaps one of the most challenging aspects of the project is the design of an appropriate mechanism to implement SLB. There are many SLB vendors and products with their own diverse installation methods, which only exasperates the decision making process at the design level. The basic idea of SLB is to make many servers appear as one to an end user by distributing the traffic load to multiple servers. There are many different ways of implementing SLB in a network; however most of them fall into two methods: Bridge-path and route-path. In the bridge-path method, the load balancer is in the layer 2 path of traffic bound for the Internet, acting as a bridge between two separate networks. In the route-path method, the load balancer is in the layer 3 path of outbound server traffic and is the server's default route. This last method is the only one that allows having the server protected in a private network. Of those two, route-path remains the better candidate.

There are many algorithms to define the load distribution policy. The algorithms used as load balancing polices in this experimental study are:

1. Round Robin, where the server alternates requests among a list of given servers.
2. A more sophisticated server load, where the Director alternates servers based on current server load.
3. Finally, a policy based on number of connections; in this case the director sends the requests based on the number of existing active connections.

In this process, confidentiality and privacy are important issues that are left as a responsibility to each participating institution, where de-identification is performed onsite prior to entering the data in this multisite web design. Each participating institution is therefore asked to remove the personal information from the data set, leaving only the relevant clinical data and medical images. The repository site does however provide elemental network security based on password protected access and a classical fire wall configuration.

4. Objectives

Preliminary research work involved a simulation model that pursued the following objectives:

1. Provide the site with fast response time even under highly-traffic sites
2. Provide high fault tolerance through server redundancy, which meant that even if an application server goes down, the site will remain operational.
3. Achieve better site performance, which can be translated into better user experience, and with faster query results.
4. Maintain the design at a reasonable cost
5. Seek high scalability
6. Attain high reliability

5. Methodology

In order to achieve the above stated objectives, the following initial steps were carried out:

1. Selection of tasks, applications and profiles to emulate the platform operation. This step incorporated the following aspects:

   a. Definition of the volume of data to be handled by the system. In other words, requirements needed to be met to contend with - the average size of a file containing MRI data - the number of files needed for a data set per subject - the average number of datasets needed per subject - the total number of patients the system can accommodate.

   b. Estimation of the number of data request received per hour or per day.

2. Selection of a proper load balancing scheme. This step is in fact a performance analysis weighing advantages and drawbacks of available load balancing polices.

3. Optimization of system response and scalability. This step is based on the simulation results, optimizing as a consequence the hardware configuration and software performance for the average situation.

The design procedure involved the following steps:

1. Define data requirement
   a. Database structure and size
   b. Expected number of queries
   c. Growth rate
2. Select the software solution to handle the problem
   a. Select either 32 or 64 bit Operating System solution
   b. Choose possible platforms
      i. LAMP – Linux /Apache/Mysql/PHP
      ii. IIS – Windows 2003/ASPX/MS_SQL
      iii. TOMCAT – Apache /Java Server Page (JSP) in Java 2 Platform Enterprise Edition (J2EE)
3. Select the hardware models needed
   a. Load balancer

This is a dedicated server running NAGIOS – Nagios is an open source service and network monitoring program that can be configured as a director [Nag05]. It can implement several polices for load balancing, the common metrics to perform SLB are per server alternation servers (round robin), per number of connections, and per current load [Car99]. An alternative to this mechanism is to use a dedicated load Balancer per hardware, such as the Big IP 5, due to the excessive cost of this hardware, that alternative is not considered.

b. Web server , database server, workstations and storage

4. Simulate using OPNET
5. Select the optimal configuration
6. Construct a prototype site at small scale based on the simulation results

5.1 The Architecture

Three architectures were suggested initially. Figure 1 depicts these proposed architectures. Those architectures were referred as:

a. Basic Architecture, where a single server takes care of all the requests, performing the functions of web server plus database server. The storage is internal.
b. Enhanced architecture. In this configuration, one server takes care of the load but the storage is external. This provides a room to scale in the future.
c. Advanced Architecture. In this configuration Load Balancing scheme is added. There are two servers providing reliability, scalability and fault tolerance to the system

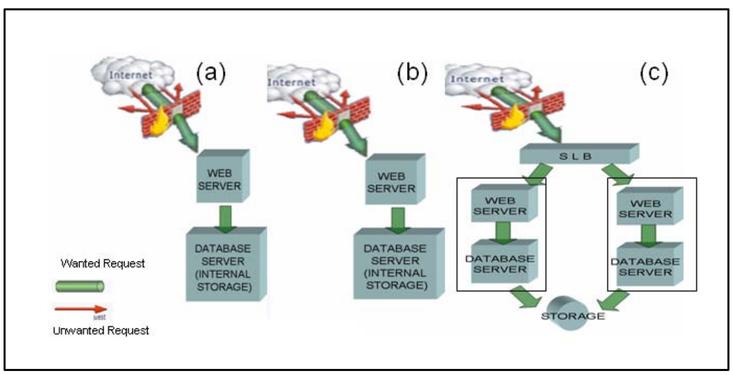

Figure 1: a) Basic Architecture, b) Enhanced architecture, c) Advanced Architecture

6. Simulation

6.1 Model

The OPNET modeler 10.0, a graphical simulation environment tailored for communicating networks, was selected as a simulation environment. OPNET features include: graphical specification of models, a discrete event simulation kernel, and integrated data analysis tools (OPNET, 2005).

A model consisting of 10 sites connected to MRI repository servers, the full name of the institutions and the number of workstation in each site is shown in table 1. The actual OPNET model is as shown in Figure 2. Two main scenarios were used, one without any load balancing scheme and one with load balancing scheme. This latter scenario was simulated using three different load balancing policies: per number of connections, Round Robin and per load.

Table 1: Distribution of Workstation and Applications in the System.

Institutions	**Number of Stations performing DB Entry**	**Number of Stations performing DB Query**
MCH: Miami Children's Hospital, Miami, USA	1	2
GOSHC: Great Ormond Street Hospital for Children, London	1	2
RCH: Royal Children's Hospital, Melbourne, Australia	1	2
HSC: Hospital for Sick Children, Toronto, Canada	1	2
CMH: Children's Memorial Hospital, Chicago, USA	1	2
SRCMC: Scottish Rite Children's Medical Center, Atlanta	1	2
CCF: Cleveland Clinic Foundation, Cleveland, USA	1	2
CHOP: Children's Hospital of Philadelphia, Philadelphia, USA	1	2
SLCH: St. Louis Children's Hospital, St Louis, USA	1	2
CNMC: Children's National Medical Center, Washington, USA	1	2
Total	10	20

The following assumptions were made during the simulation:

1. Each site is connected to the internet cloud using a T1 link. Bandwidth is assumed to be symmetric and equivalent for all sites.
2. Each site has 3 workstations connected. One computer uses a profile deploying medium database entry operations, while others use profiles performing intensive web browsing and intensive data base querying applications in parallel. Figure 2.c depicts the subnet used at each medical institution.
3. Three applications were used as traffic generator for the simulation. In each subnet, one station performed database entry operation at medium load and one station performed DB query as an intensive task. The configurations of the applications are illustrated in Figures 3 and 4.

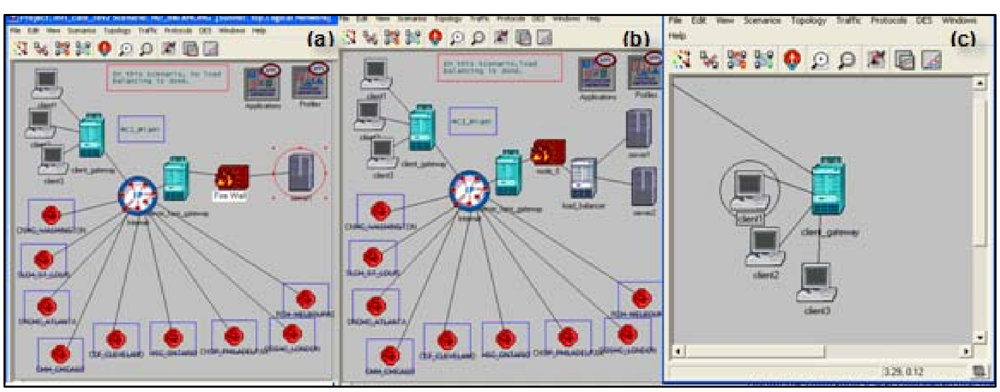

Figure 2: OPNET Models. (a) No load balancing scenario, a model for basic architecture , (b) Load balancing scenario, a model for advanced architecture) , and (c) Subnet model

4. The model used for the server object has a single processor running at 3.0 GHz, the model for the links in the intranet is Ethernet running at 100 Mbps, while the link between gateways and the internet cloud are lines DS1 at 1.54 Mbps.

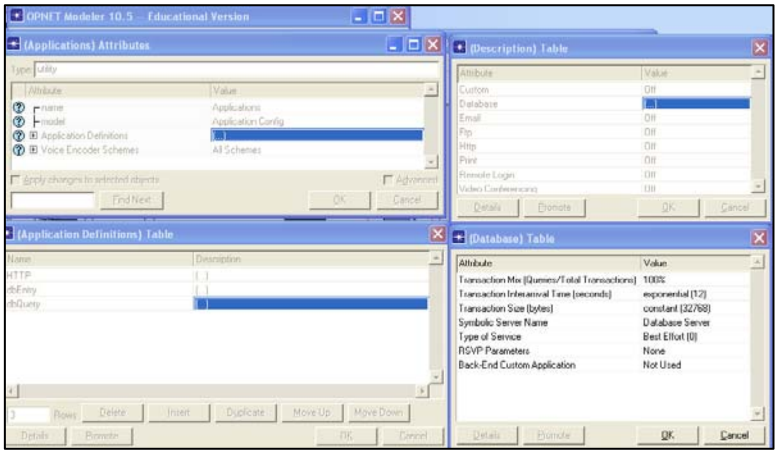

Figure 3: Data Base operation definition graphical interface.

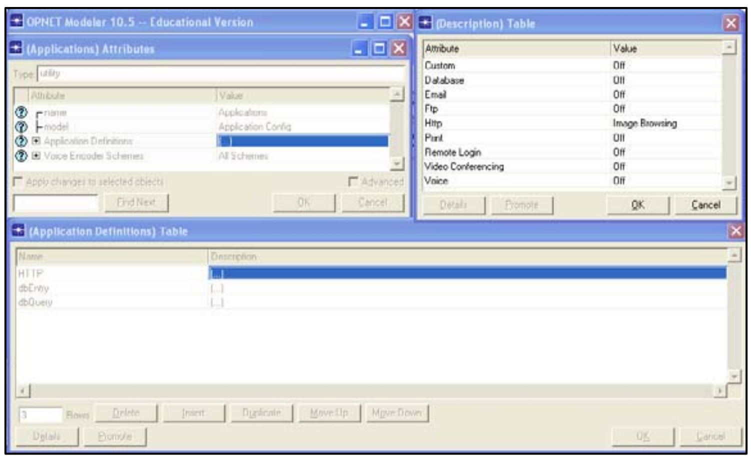

Figure 4: Browsing operation definition graphical interface.

6.2 Results

OPNET was able to provide different statistical results of the network operation. Among the several simulation results available, the following were chosen: the CPU utilization of the server, the server throughput, the link usage between the server and the nearby gateway, the load of the server and its throughput.

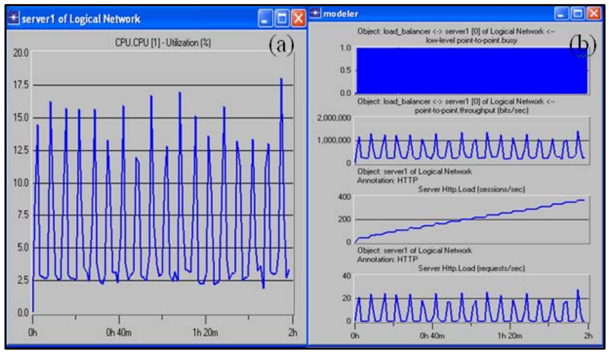

Figure 5: Results of network performance without load balancing: (a) CPU utilization, and (b) Link utilization, server throughput, server load and server load request

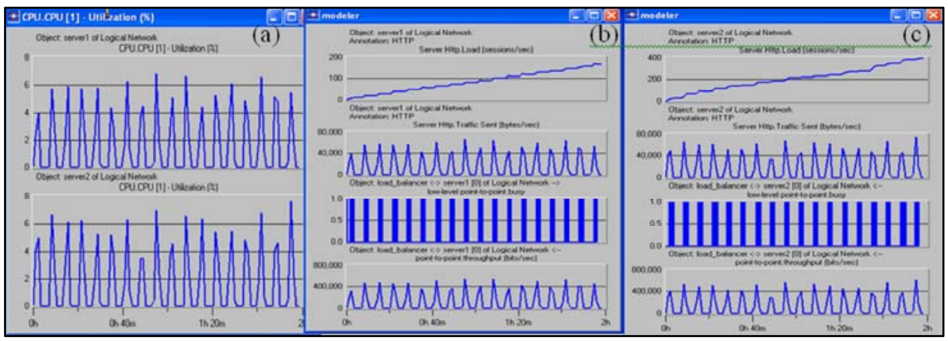

Figure 6: Results of network performance with load balancing using number of connections as SLB policy : (a) CPU utilization , (b) server load, server traffic, link utilization, and server throughput for server 1, and (c) server load, server traffic, link utilization, and server throughput for server 2.

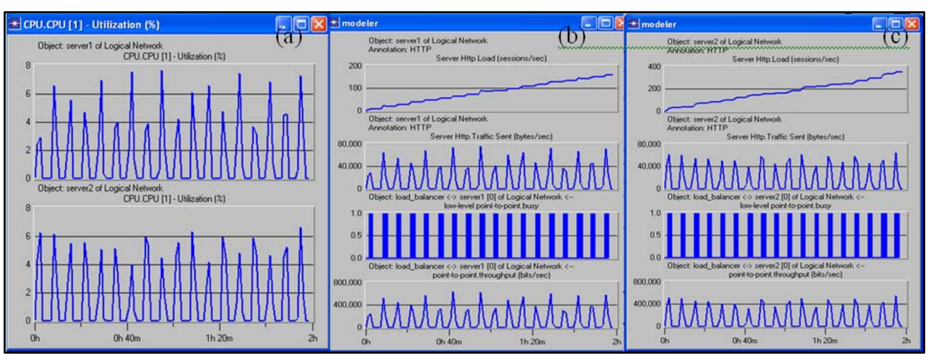

Figure 7: Simulation Results of network performance with load balancing using Round Robin as SLB policy: (a) CPU utilization, (b) server load, server traffic, link utilization, and server throughput for server 1, and (c) server load, server traffic, link utilization, and server throughput for server 2.

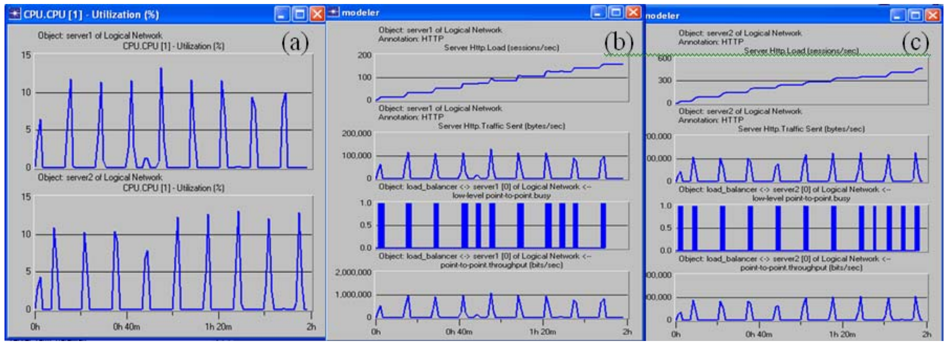

Figure 8: Results of network performance with load balancing using server load as SLB policy: (a) CPU utilization, (b) server load, server traffic, link utilization, and server throughput for server 1, and (c) server load, server traffic, link utilization, and server throughput for server 2.

The information collected from those selected criteria provided relevant data to be able to decide whether to select or not the load balancing and its associated policy.

7. Discussion

During the simulation, the IP background traffic in the network was not considered. A heavy IP background traffic can cause early congestion and system degradation. Attention was thus focused on the server side; no analysis was done on the client site. No change was done on the application definition during the whole simulation process.

The simulation process was executed in periods of 2 and 4 hours operation. A more extensive analysis period remains to be observed. The results of the simulation without load balancing shows a constant utilization of the link status to the server (see low-level-point to point busy status in Figure 5.b) without any chance to increase the server throughput (close to 1 Mbps). Although the simulation showed a server CPU utilization inferior to the 20%, the link utilization, in the point to point connection between the server and the attachment point to the internet, prevents reaching higher CPU usage (in this layout the load balancer object is just a pass-through object).

For the SLB based on number of connection a highly symmetric CPU utilization was achieved. However, the total throughput of the network servers is lower than the one with a single server (see Figure 6).

The Round Robin paradigm in SLB operation showed the worst results. The CPU utilization was asymmetric and the throughput was low. However, the server throughput and the link point-to-point status were similar to the ones exhibited by the SLB based on number of connections (See Figure 7).

Results shown in Figure 8 depict a better server throughput but slightly asymmetric CPU utilization between the servers. Moreover, the link point-to-point status was better than the ones experienced by the other two SLB paradigms.

8. Conclusion

Based on the results observed during the simulation it is deemed sensible to use load balancer in the configuration, although the CPU utilization is not so high during one server operation, the link utilization was saturated and a higher risk of failure is more probable. Moreover, there is almost no margin to allow a load growth. Using SLB the link becomes more available and the utilization of each server is lower as seen in the simulation results shown in Figures 6, 7 and 8. Since both servers are available the site has a high fault tolerance and a higher scalability as well.

The best performance is achieved when using SLB with server load as policy (see Figure 8), although the CPU utilization is not symmetric, the throughput achieved by the servers is the highest for the SBL scenarios. Furthermore, the link busy status is relatively low with a CPU utilization with peaks slightly superior to 10%, which provides a good margin for load growth as shown in Figures 8.b and 8.c.

The simulated approach presented in this paper will help in the final implementation of the actual multisite for image and medical information storage. The purpose of achieving a scalable, reliable and fault tolerant site at low cost can therefore be met.

References

Cardellini, V., Colajanni, V. M., and Yu, P.S. (1999). " Redirection Algorithms for Load Sharing in Distributed Web-server Systems". In the *Proceedings of IEEE 19th Int. Conf. on Distributed Computing Systems (ICDCS'99)*, Austin, TX.

Chu, W., Cardenas, A., and Taira, R. (1995). "KMED: A Knowledge Based Multimedia Medical Distributed Database System". *Information Systems*, Vol 20, issue 2, pp 75-96.

Chen, Zhe, Yu, X., and feng, D. (2001)."A Telemedicine System over Internet", *Conference in Research and Practice in Information*, Vol 2.

Lin, y., Feng, D., Cai, T. W. (2001). "A web-based Collaborative System for Medical Image Analysis and Diagnosis" , Australian Computer Society, Worksop on Visual information Processing, *Conference on Research and Practice in Information Technology*, Vol 2.

Lin, C., Reng, J., et all. (1998). "A Unified Multimedia Database System to Support Telemedicine", *IEEE Transactions on Information Technology in Biomedicine*, Vol2, No 3.

Munch, H., Engelman, U., et all. (2003). "Web Based distribution of radiological images from PACS to EPR", *International Congress Series 1256*, pp 873-879.

NAGIOS manual. (2005). , *http://www.nagios.org 09/27/05*. (Date accessed)

OPNET Modeller, OPNET Technologies Inc. (2005) *http://www.opnet.com/services/university/* .
08/21/05. (Date accessed)

Authorization and Disclaimer

Acknowledgements

This work is supported by the American Epilepsy Society and the National Science Foundation.