Operational Departmentwide Picture Archiving Communication System Analysis Using Discrete Event-Driven Block-Oriented Network Simulation

Brent K. Stewart

The accurate prediction of image throughput is a critical issue in planning for and acquisition of any successful picture archiving and communication system (PACS). Simulation plays an important role in this effort. The PACS image management chain is decomposed into eight subsystems. These subsystems include network transfers over three different networks and five software programs and/or queueing structures. This decomposition is used to create a simulation model that was effectuated using commercially available block-oriented network simulation software. From the PACS database, the traffic generation patterns of the imaging modality devices are used to drive the simulation. The simulation models the image file flow through the PACS for a 24-hour period. The behavior of the simulated traffic generators agreed well with the values derived from the PACS database. The mean delay for the simulated PACS is found to be 225 ± 59 seconds. The delay time was found to vary during the simulated 24-hour cycle in a consistent manner with observations. This simulation provides estimates on what a radiological department can expect from a PACS in terms of throughput, utilization, and delay. The block-oriented network simulator (BONeS, Comdisco Systems Inc, Foster City, CA) simulation model of the modeled PACS is highly accurate. The models for the imaging modality traffic sources are validated with a high degree of accuracy. The simulation model allows for the study of what happens to the delay time under various loads. In this simulation, the reformatting process was determined to be the bottleneck causing a large increase in delay time under heavy loads.

KEY WORDS: communication network, picture archiving and communication system, queuing model, simulation, system analysis.

THE ACCURATE prediction of image throughput is a critical issue in planning for and acquisition of any successful picture archiving and communication system (PACS). Bottlenecks or design flaws can render an expensive PACS implementation useless. This article presents a method for accurately predicting and measuring image throughput of a PACS design (Fig 1).

Performance evaluations and trade-off analyses are the central issues in the design of communication networks.[1] To estimate system performance, one may do one of the following:

1. Perform an ex post facto analysis on an already existing system. This "build first and ask questions later" philosophy most often leads to unwise purchases and unhappy users.
2. Make a simple projection by scaling up from existing experience. However, the behavior of most communication systems is not what one would intuitively expect. When there exist shared facilities, a system's performance typically responds in an exponential rather than linear fashion to increases in facility demand.
3. Develop an analytical model based on queuing theory[2] or other parameterized deterministic model. Even through models based on queuing theory can provide a good fit to some "real world" problems, one must make some simplifying assumptions in the derivation of these equations.
4. Program and run a simulation model.[3-6] Given a sufficiently powerful and flexible simulation programming language or software package, one can model reality in as much detail as is required and avoid making the assumptions required of queuing theory.

Modeling and simulation are used for capacity planning (to meet specific performance criterion), for the analysis of various trade-offs and "what if" scenarios, and in the evaluation of alternative architectures (design iterations). It also aids in configuration optimization and extended connectivity options, as in assessing the impact of adding teleradiology to an existing PACS.

Simulation is a series of techniques for evalu-

From the Department of Radiological Sciences, University of California at Los Angeles, School of Medicine, Los Angeles, CA.

Address reprint requests to Brent K. Stewart, PhD, University of California at Los Angeles, Department of Radiological Sciences, 10833 Le Conte Ave, Los Angeles, CA 90024-1721.

0897-1889/93/0602-0005$03.00/0

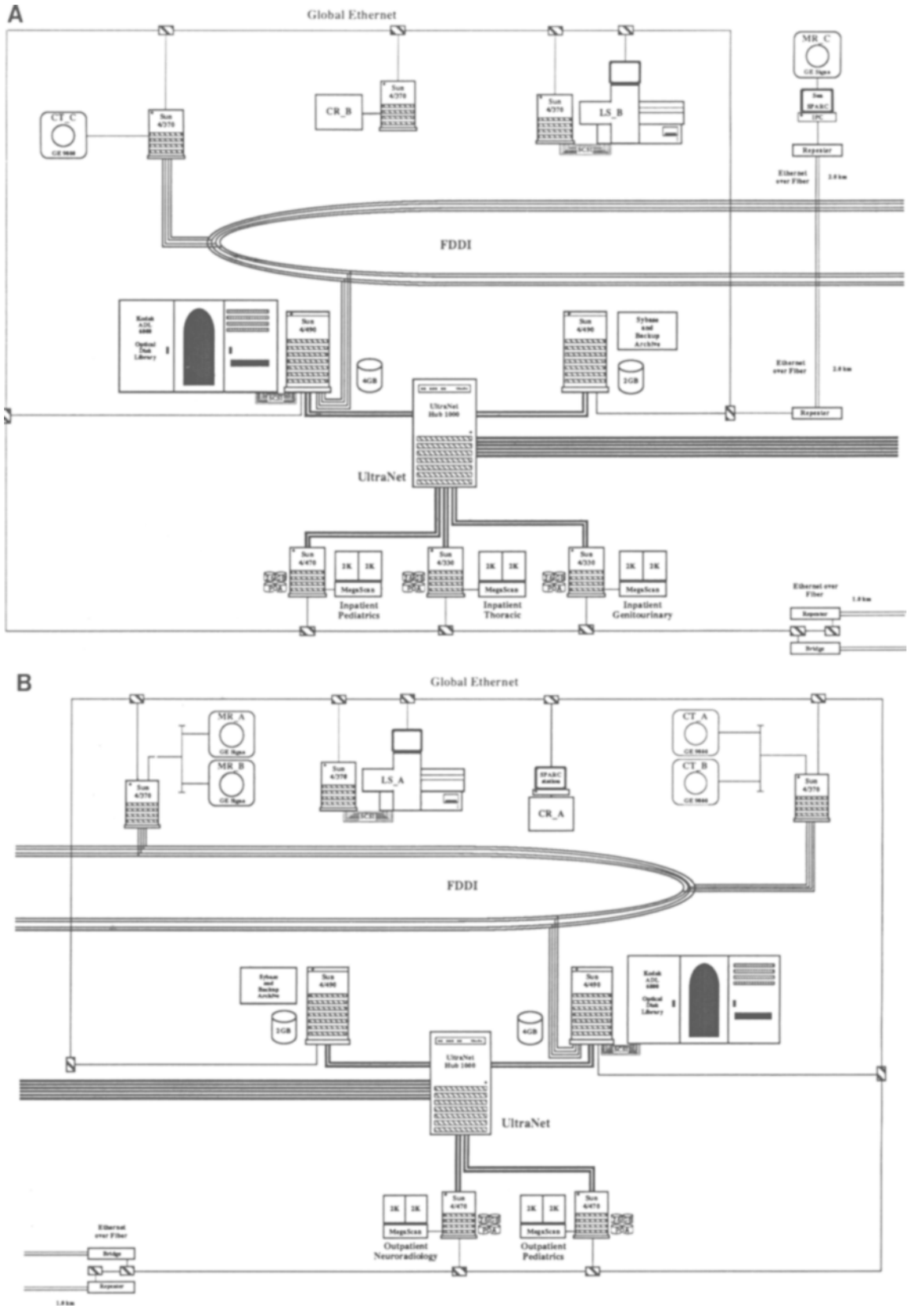

Fig 1. Diagram of the PACS showing the computer systems and networks involved. It is this configuration of hardware that is modeled in the simulation. *A* corresponds to the Center for Health Sciences and the connection to the Advanced Imaging Center. *B* corresponds to the recently completed Medical Plaza Outpatient Facility.

ating the performance of a system or process by developing a parameterized model of the system or process and then iterating the model through a range of values for one or more of the parameters to observe the output behavior of the model. There are several approaches available for simulation: (1) general programming languages such as C and FORTRAN; (2) simulation languages such as SIMSCRIPT,[7] SIMULA,[8] and SLAM[9]; (3) parameterized deterministic mathematical models such as finite state models,[10] Markov chains,[11] Petri Nets,[12] or queuing theory[13]; (4) event-driven parameterized models such as the generalized local area network (LAN) analysis system (GLAS)[14]; and (5) graphical, icon-based simulation environments, of which several commercial products now exist at various levels of complexity and price.[15]

A common problem in simulation is deciding what level of detail is appropriate for the model at hand. Simulation allows a system to be studied in greater detail than analytical modeling in which the development of tractable mathematical models with closed form solutions is generally impossible without simplifications and assumptions. In a simulation model, the level of detail is limited only by the time available for simulation development and the patience of the developer. A very detailed simulation requires more time to develop, debug, and run. A common mistake is to take the detailed approach when a high-level approach will do and contrariwise. The goal then in regards to selection of the level of detail is to select the optimum level of abstraction which is low enough to investigate performance issues, yet high enough to allow efficient simulations. It is best to start off with a high-level model, obtain preliminary results, study sensitivities, and then introduce detail in the parts of the model that have the greatest impact on the simulation output.

THE BLOCK-ORIENTED NETWORK SIMULATOR

The steps in the modeling and simulation process are: (1) determine what questions the simulation is to answer, (2) define performance metrics, (3) perform measurements, (4) create the simulation model, (5) execute simulation, (6) analyze simulation result, (7) verify the model, and (8) use the reduced simulation output for decision support.

BONeS provides an interactive graphical environment for discrete-event simulation-based communication network design and analysis.[16] In the BONeS environment, the network model is specified in terms of the network topology (a hierarchical data-flow block diagram), the functional elements that make up the topology (building blocks), and the data structures that traverse the topology (files, traffic, packets, messages, etc). BONeS provides primitive building blocks (modules) with which to initiate this process. These primitives perform simple functions, such as a first-in-first-out (FIFO) queue. Using these primitives, higher level building blocks in the BONeS model library and user-defined modules, a hierarchical network model is constructed. One may also write model components in the C programming language and incorporate them into the BONeS modeling environment.

BONeS translates the network model into a C language program and executes an event-driven Monte Carlo simulation. Event-driven[1] means that the BONeS executable program actions are triggered by the occurrence of specific events, such as the arrival of a packet or message. This makes for efficient use of computer processing time, in contrast to time-driven simulations in which at each tick of the simulation clock, all modules are queried for input and output. However, time-driven simulations are well-suited for the waveform level simulation of integrated circuit behavior and for signal processing applications.[17] Monte Carlo[5] means that the behavior of the network is simulated by randomly varying the value of intervening variables which control the operation of simulation models over a length of time, to ensure that the states of the intervening variable are properly represented. During simulation, probes inserted at strategic positions throughout the model extract and reduce desired data for throughout, delay time, etc. After the simulation execution, several BONeS analysis and presentation tools are available to process the data gathered by probes.

The main components of the BONeS software are a data structure editor (DSE), a block diagram editor, a simulation manager, and the postprocessing facility (Fig 2). In addition to the above four user-interactive components of the

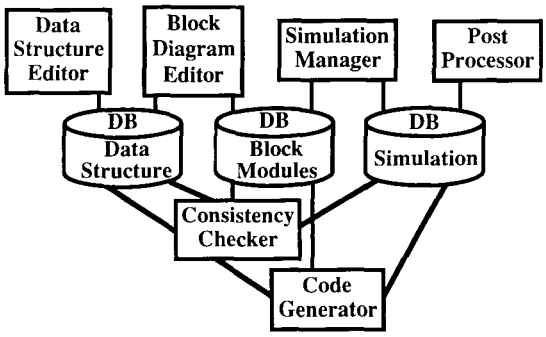

Fig 2. Principal external and internal components of the BONeS software package. The user interacts with the top four program modules: data structure editor, block diagram editor, simulation manager and postprocessor. Below this level are internal databases, checkers and compilers that the user never need be concerned with. No programming is required to design, construct and execute even very detailed simulations, but the serious user can create customized primitive modules through the C programming language.

BONeS software, there are other internal components (Fig 2): internal integrated database managers, consistency checkers, code generators and libraries of data structures, block diagrams, protocol functions and LAN models.

The DSE is used to create, edit, document, and store data structures. The DSE is composed of the type hierarchy graph (THG) window and the data structure format window. The THG is a visual representation of the hierarchy existing in the data structures. The data structures are defined to meet the requirements of the simulation and need not necessarily duplicate actual packet structures in the network being simulated. Hierarchy and encapsulation in the DSE provides an easy and efficient means of modeling layered architectures as in network protocol stacks (eg, Transmission Control Protocol/Internet Protocol, Open Systems Interconnection, and Systems Network Architecture). It is the data structures created using the DSE that are generated by various traffic sources and processing blocks and transmitted though the block diagram network during the simulation.

The block diagram editor is used to create, edit, document, and store the network of block diagrams which comprise the simulation model. BONeS uses a hierarchical data flow block diagram modeling paradigm for networks. Primitive modules serve as the starting point for this hierarchy. Primitive modules have inputs and outputs, but unlike other modules they cannot be broken down into submodules. The functionality of primitive modules is derived from hand-coded C language source code rather than by the presence and arrangement of submodules, as is the case for all other types of modules. Thus, blocks represent C programming language functions. One of the most important characteristics of BONeS is the hierarchical relationship among modules. New modules are typically built from existing modules. The new modules can be used to build other modules. This hierarchical structure allows a system problem to be broken down into smaller, conceptually contained modules. The inputs and outputs of these small modules can be connected graphically to form the model of the system being simulated. This connected network of modules comprises a C language program which is then compiled using the simulation manager. No programming is involved, simply the graphical connection of modules and primitives, many of which already exist in the BONeS model libraries.

The simulation manager generates, submits, and monitors the execution of simulation programs. It is at this point in the simulation that simulation-level variable parameter values are set (eg, channel rates, queue buffer sizes, traffic generator interarrival times, etc). Probes, also consisting of modules, are placed at various locations throughout the simulation model to gather and reduce desired data, which are written to a magnetic disk file while the simulation is executed. After a variety of error and consistency checks, the simulation manager compiles the hierarchical network model into a C program executable object. This executable, an event-driven Monte Carlo simulation, can then be run locally or elsewhere on a different computer on any attached LAN.

The postprocessor allows for analysis and display of simulation results. Multiple simulations of the same model are grouped together in the databases so that statistical operations (eg, averages and confidence intervals) for various performance metrics (eg, delay time or throughput) can be computed and plotted as functions of simulation parameters (eg, mean interarrival time). The postprocessor graphs can be printed or the data values can be output to a magnetic

disk file for export to other plotting packages or spread sheets.

PACS PRINCIPAL TASK DECOMPOSITION

To create the simulation model of the planned or implemented PACS (eg, Fig 1), it must first be decomposed into principal tasks (Ts) (Fig 3). These principal Ts involve the transfer of image files from the image source (image modality) to image sinks (optical archive or image display workstation):

(T1) Transfer the newly generated image data from the imaging modality computer to the image acquisition computer (Ethernet).

(T2) Reformat the image data from a proprietary vendor format into a standard image format, such as American College of Radiology–National Electrical Manufacturers Association (ACR-NEMA),[18,19] by an acquistion and reformatting computer.

(T3) Transfer the image data from the acquisition/reformatting computer to the image archive computer (fiber distributed data interface [FDDI] or Ethernet).

(T4) Update a relational database management system over the network (Ethernet or UltraNet).

(T5) Perform image processing (IP) in the image archive computer: rotation of computed radiography (CR) images (if required) and optimal gray scale lookup table calculation for CR, computed tomography (CT), and magnetic resonance images (MRIs).

(T6) Request that the image be archived (request placed in a queue) to an optical disk library.

(T7) Transfer image from the image archive computer to a designated image display workstation (UltraNet or Ethernet).

(T8) Update the local database on the image display station, separate the image header from the image data, and store the image data on a parallel disk array, ready for review.

TRAFFIC GENERATOR CHARACTERIZATION

The traffic generator modules (image sources) representing the various imaging modalities drive the simulation. Except for some routing algorithms, the remainder of the simulation modules transfer image files passively, handing images from one process or network to another. Thus, great attention was paid in characterizing the image file length, image display station destination, and temporal frequency distributions of each modality for use in the appropriate traffic generator modules. Data for the characterization of the image file length, destination, and temporal frequency were extracted from our PACS relational database management system (Sybase; Sybase Inc, Emeryville, CA) for a period of 4 months (May through August 1991).

Both constant and variable file lengths are observed in the PACS database. Images from CR and laser scanned (LS) film digitizers are predominantly of one size: 8 Mbytes (2,048 × 2,048 pixel × 12 bits; 16-bit word boundaries). CT and MRI file lengths depend on the imaging sequence used and vary signifi-

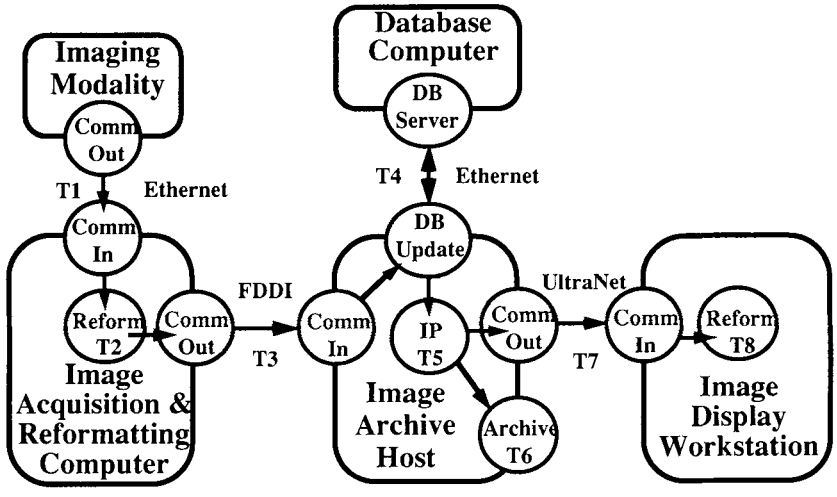

Fig 3. PACS principal task decomposition. The PACS imaging chain has been divided into eight logical tasks. This diagram follows the logical flow of images from the imaging modality clear through to the image display workstation. During this trek, the images pass through three different networks, an image acquisition and reformatting computer, and an image archive host.

cantly from one examination to another. The CT and MRI file length distributions are closely exponential in character.

Image file destination refers to which image display station(s) the image will ultimately be routed to. This is extracted from the database as a section code based on the anatomy being imaged and whether the patient is an inpatient or outpatient. The five workstations currently operating in the department are genitourinary inpatient, neuroradiology outpatient, pediatric inpatient, pediatric outpatient, and thoracic inpatient.

Because the time interval between the generation (creation) of images varies drastically over a 24-hour period, a solitary mean intercreation interval will not model peak clinical traffic patterns well. To model the variation in image generation frequency during a 24-hour period, this large period is divided into 1-hour periods from which one can calculate the hourly mean intercreation intervals.

From multiple database tables in the global PACS database,[20,21] the following information was extracted for each imaging modality unit using structured query language (SQL) queries: (1) file lengths, (2) display station destination, and (3) the time difference between image creation (intercreation interval). This data was used to determine parameters necessary for the image traffic generator modules that will be described. Table 1 gives the measured mean, maximum, and minimum file lengths for the three CT and three MRI scanners. The mean file lengths in Table 1 are used in the simulation.

Because the values obtained for the intercreation interval during each 1-hour period included some large values that skewed the mean, a logarithmic transform[22] was used to make the distribution more normal. Table 2 shows the result that the logarithmic transform has on the mean intercreation intervals. The general result is a decrease in the mean, especially so during the very early morning hours when the mean intercreation interval is large. The logarithmically transformed data yields more consistent validation results.

Table 1. The Mean, Maximum and Minimum File Lengths for the CT and MRI Scanners

Imaging Modality	File Size (Mbytes)		
	Mean ± SD	Maximum	Minimum
CT_A	10.23 ± 1.09	64.06	0.50
CT_B	9.77 ± 2.62	48.55	0.50
CT_C	11.49 ± 2.65	64.06	0.50
MR_A	2.26 ± 0.26	7.53	0.13
MR_B	2.05 ± 0.20	15.56	0.25
MR_C	2.62 ± 1.34	18.57	0.13

NOTE: Data was derived from the PACS database for the time period of May through August 1991. The mean file lengths in this table are used in the simulation.

Table 2. The Mean Intercreation Intervals Calculated From the PACS Database for One of the CT Imaging Devices

Hour Time Slots	Mean Intercreation Interval for CT_B (sec)		
	Log Transform Mean	No Transform Mean	No Transform SD
0:00	4,074.6	4,320	±1,642
1:00	4,706.7	6,276	±4,542
2:00	7,894.9	7,674	±6,457
3:00	3,361.4	8,902	±6,739
4:00	7,097	17,028	±4,618
5:00	7,652	8,513	±4,682
6:00	7,591.7	7,140	±1,714
7:00	3,840	5,700	±1,613
8:00	4,410.7	4,423	±2,150
9:00	4,727	4,143	±2,923
10:00	2,706.8	3,028	±1,618
11:00	2,901.3	3,540	±2,085
12:00	2,486	3,317	±2,041
13:00	3,055.2	3,567	±1,625
14:00	2,991.2	3,477	±1,588
15:00	3,142.4	2,872	±1,838
16:00	2,405.9	3,266	±2,320
17:00	2,364.5	3,021	±1,824
18:00	3,127.6	3,773.3	±2,257
19:00	2,601.5	3,430	±2,203
20:00	2,929.2	3,311	±2,293
21:00	3,167	3,690	±2,328
22:00	2,649	3,116	±2,945
23:00	2,310.7	3,673	±3,246

NOTE: The entries in the "no transform" columns are the mean values calculated by measuring the time difference between image transmission from the respective imaging modalities during each 1-hour period. A logarithmic transform was applied in an attempt to "normalize" the distributions. The "log transform" data column is that used in the simulation for the CT_B traffic generator.

SIMULATION MODEL

The departmentwide PACS in the department includes as image sources three MRI systems, three CT Systems, two CR systems and two film digitizers (LS), located throughout three separate buildings.[23,24] For networks,[25] the PACS has a logically global Ethernet, one FDDI ring, and two UltraNet[26,27] hubs. In

addition, the PACS has two image archive servers that host two optical disk archives and five image display stations.

This PACS model is a large-scale, multi-input, multi-output queuing model with variable and fixed delays mimicking network file transfer and file processing times. The simulation models the image file flow through the PACS for a 24-hour period. The top-level modules comprising the simulation model are shown in Fig 4 based on the PACS diagram in Fig 1.

Data structures that represent image files created by the traffic generators (eg, CT_A) typically pass through Ethernet connections to reformatting processes running on the various acquisition and reformatting computers. Next, the data structures are transferred to the archive computers through either Ethernet for FDDI. The archive computer is composed of the following modules: routing switch, image processing, database transactions, and an archiving process. UltraNet is used to transmit images between archive computers. After passing through the archive computer, the data structure is transmitted through the UltraNet module and into the display station module.

SPECIFICATION OF SIMULATION PARAMETERS

Traffic Generators

It is the responsibility of the traffic generator modules to determine what length file to create, what the image display station destination should be, and when to send an image file. Currently, only fixed length files and exponentially distributed variable file lengths are supported. Fixed length files are used for CR and LS film images that are always 8 Mbytes. The CT and MRI file length distributions are more closely exponen-

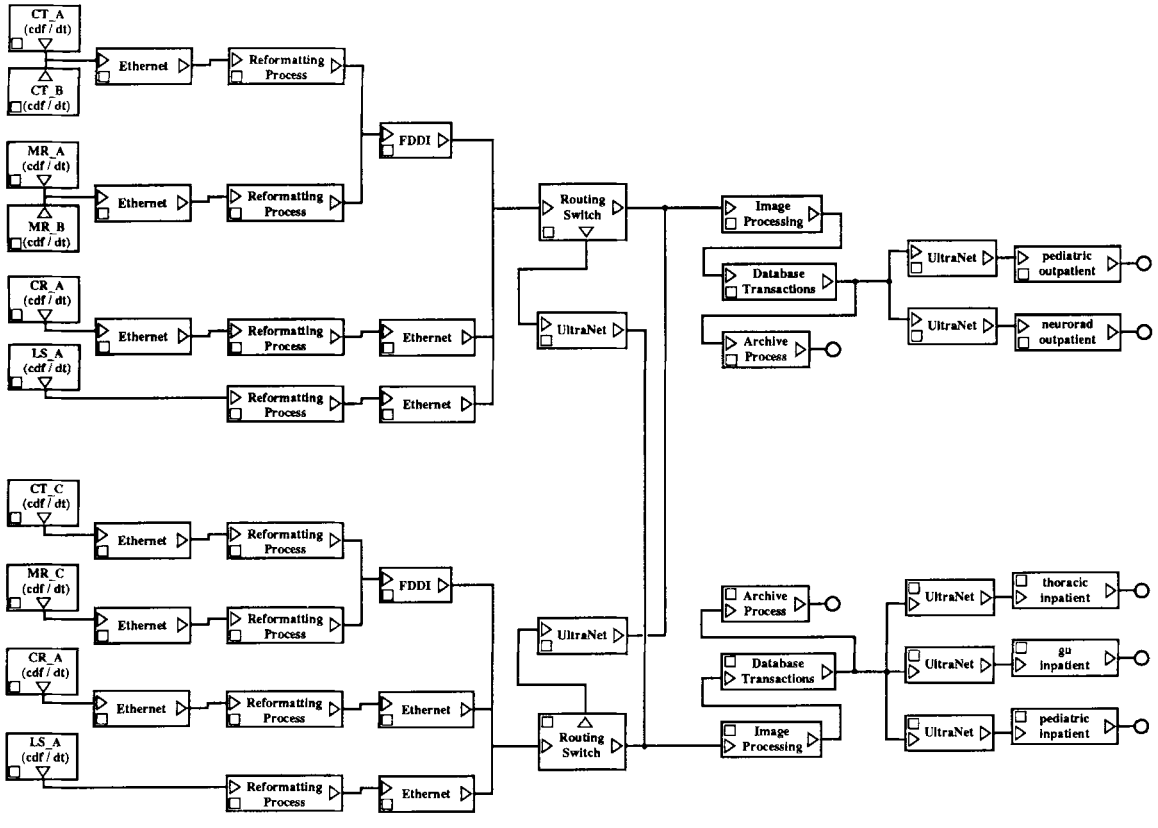

Fig 4. Simulation block diagram model for the PACS derived from the principal task decomposition given in Fig 3 and from the topology given in Fig 1. On the extreme left are the ten traffic generators representing the imaging modalities. Data flows in the system towards the right side where the five image display workstations reside (eg, mega1.ped2—pediatric outpatient). The top half of this block diagram represents Fig 1B, whereas the bottom half of this block diagram represents Fig 1A. The circles represent final image sinks, either optical disks or fast magnetic disk arrays.

tially distributed. In the future it will be possible to specify any arbitrary distribution function. In addition, the distribution of image display station destinations are specified using a cumulative distribution function.

The mean intercreation interval between image files is specified for each individual traffic source (eg, Table 2). The temporal distribution of image file creation is exponential, with the mean intercreation interval for each 1-hour simulation clock interval serving as the sole parameter.

Figure 5 shows the temporal and file length distribution for the generation of image files from one of the CR, CT, LS and MRI traffic generators. The computed radiography (CR_A) and laser scanned (LS_A) film traffic generators create only fixed length files of 8 Mbytes. The computed tomography (CT_B) traffic generator file lengths are exponentially distributed with a mean file length of 9.77 Mbytes. The magnetic resonance image (MR_B) traffic generator file lengths are exponentially distributed with a mean file length of 2.05 Mbytes. The time of day refers to when the image file is created. Note that a majority of images are created during the peak hours of 9 AM to 6 PM. The time-varying and stochastic exponential distribution of intercreation intervals yields the concentration and tenuousness of image creation during the 24-hour period. The displayed file length ranges correspond well with those given in Table 1.

Network Transfer and Host Processing Rates

The processing delay of a file through each module in the PACS simulation may be dependent or independent of the image file length. An example of a module where the processing delay is independent of the file length is the database transactions module. An example of a module in which the processing delay is dependent on file length is a module representing a network transfer (UltraNet) or disk access (reformatting process).

Modules in which the processing delay is independent of image file length are modeled

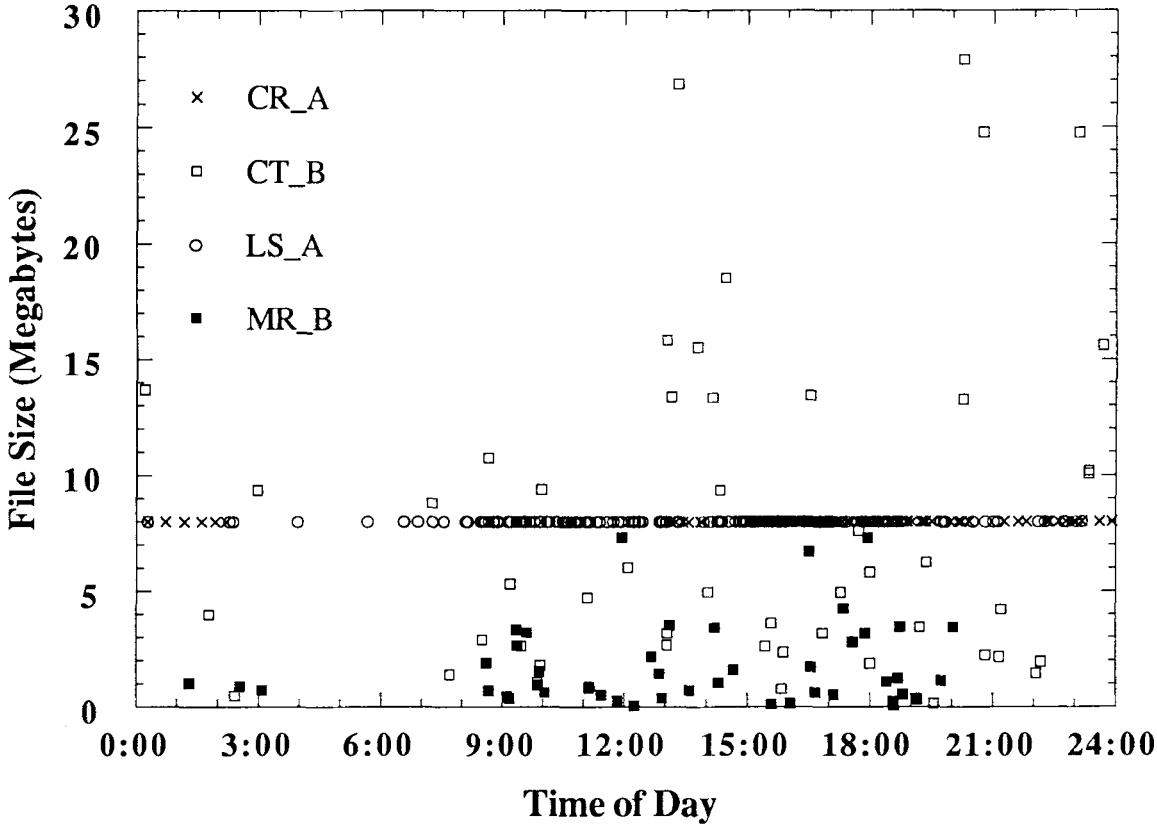

Fig 5. This graph shows the temporal and file length characteristics for four of the traffic generators in Fig 4.

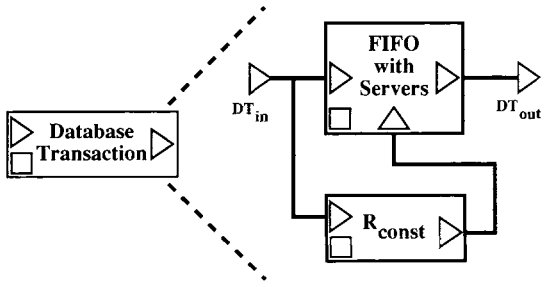

Fig 6. File size independent processing module consisting of queue and server with fixed delay. The database transaction (DT) block consists of two lower level subblocks: a FIFO queue with servers and a constant valued real number generator (R_{const}). The triangular input (DT_{in}) and triangular output (DT_{out}) correspond to the triangles in the database transaction block. The squares in the lower left corners of each block indicate that the block contains a variable that requires definition (eg, real constant).

by an FIFO queue with fixed processing delay server (Fig 6). This fixed processing block emulates well the software queues and magnetic disks of the modeled PACS.[28] Modules in which the processing delay is dependent on the image file length are modeled by a FIFO queue and modules that extract the file length and effective throughput rate of the network/disk/process (channel speed), and then they are able to divide the two (Fig 7). This value then becomes the delay time through the module. This variable delay processing block emulates well the transfer of image files, packets, and messages through a network channel. The effective throughput rates measured for the PACS

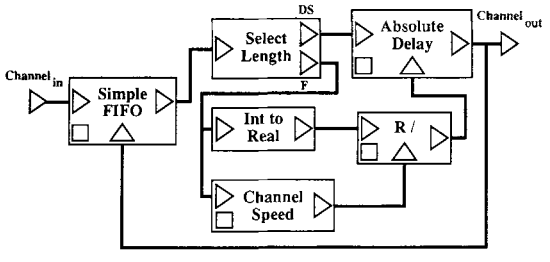

Fig 7. File size dependent channel processing module used for emulating network file transfer. The input to this block ($Channel_{in}$) enters into a FIFO queue (Simple FIFO). When the queue is empty, the entering data structure passes through to the select length block where the data structure length (number of bytes) field is copied and passed to an integer to real number converting block (Int to Real). At the same time, the channel speed (bytes/second) is fed into a real division block along with the real number of bytes and divided (bytes ÷ bytes/sec). This result is the delay time to the channel which is fed into the absolute delay block, retarding the data structure (file) until the delay time has passed and it is sent to $Channel_{out}$. When this occurs, the Simple FIFO block is notified to release the next data structure in the queue, if any.

Table 3.
The Throughput and Processing Time Parameters Used in the Simulation for the Tasks Outlined in Fig 3

Simulation Task	Throughput Rate or Processing Time
Ethernet w/SCSI disks	180 kbytes/sec
Image reformatting	80 kbytes/sec
FDDI with SCSI and IPI disks	500 kbytes/sec
Database transactions	0.1 seconds
Image processing	10 seconds
Archive process	250 kbytes/sec
UltraNet with IPI disks	1,250 kbytes/sec
Display station processing	700 kbytes/sec

NOTE: Values issued to the simulation through the BONeS simulation manager.

Abbreviations: SCSI, small computer systems interface; IPI, intelligent peripherals interface.

and used in this simulation are given in Table 3. The FIFO queues in these modules emulate the queues in the PACS software architecture and also serve to emulate the image files being stored on magnetic disk.

Simulation Probes

Several methods may be used to extract meaningful performance measures from a BONeS network simulation model. One approach is to design the data structures for the model so that fields are included whose values allow statistics to be computed. Another approach to statistical collection is to build the statistical computations into the block diagrams of interest, what is known as a probe. A probe is a simulation module that acts as a filter, taking data structures flowing through the block diagram, manipulating the data into a form that is less bulky or has additional information.

The BONeS simulation manager is used to attach probes at strategic positions within the block diagram, specifying the collection of data from the simulation. Probe modules typically provide such functions as averaging, computing higher order moments, variance reduction, or time sampling. Any data structures that pass from the output of the probe are written to a disk file that may be accessed with the BONeS postprocessor. Figure 5 and Figs 8 through 11 were plotted with data from such probes.

RESULTS

Traffic Generator Output and Validation

Figure 8 describes the cumulative megabytes generated by three of the ten traffic generators

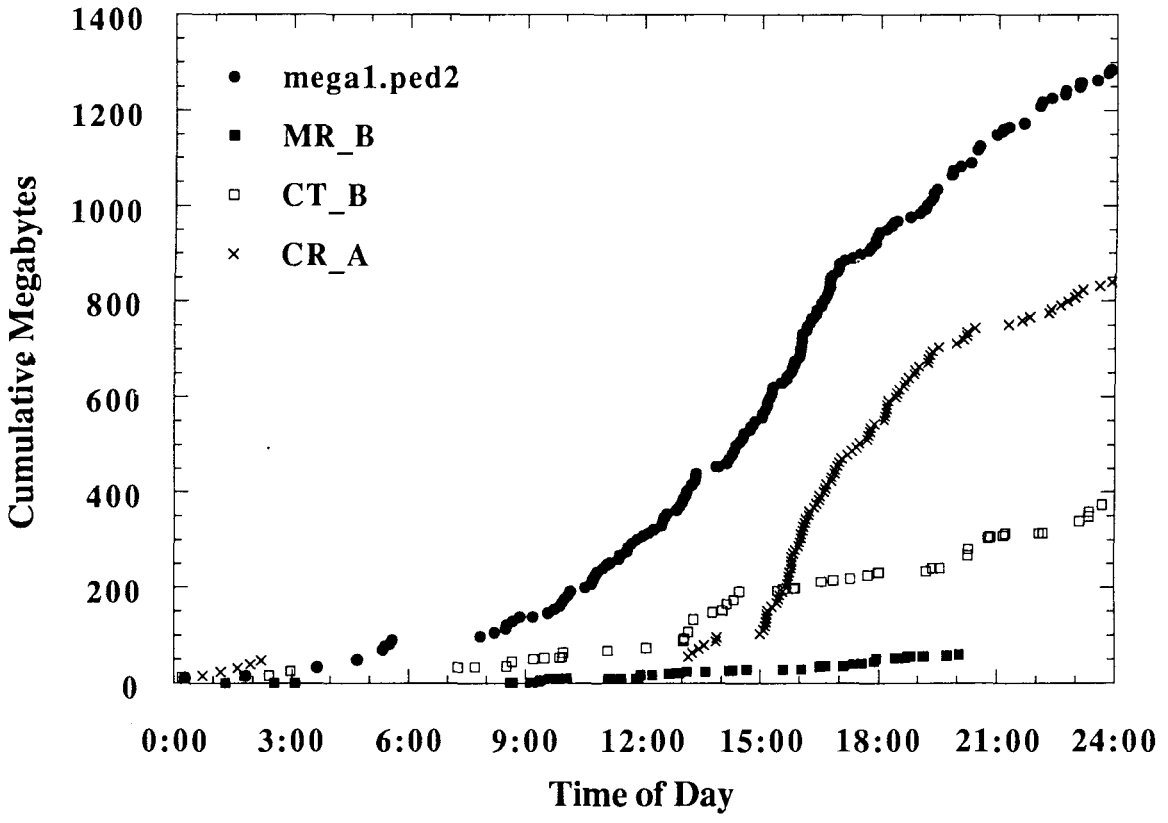

Fig 8. This graph shows the cumulative number of megabytes generated by three of the traffic generators and the cumulative megabytes arriving at a display station (mega1.ped2) versus the simulation clock.

(one CR, one CT and one MRI) versus the time of day clock. Each point in the graph indicates that a single image file has been created at a certain time (abscissa). The sum of these image file lengths gives rise to the cumulative megabytes (ordinate). Figure 8 also contains the cumulative number of megabytes arriving at the outpatient pediatric image display station (mega1.ped2) for the same 24-hour period. Note again how the 24 mean intercreation intervals specified to the traffic generators increases traffic during the clinical hours of 9 AM to 6 PM, whereas there is little traffic during the early morning hours.

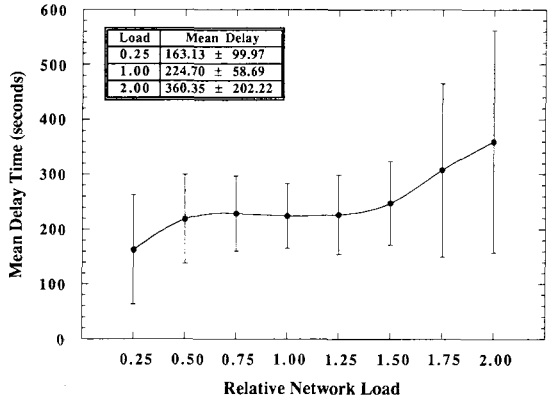

Fig 9. Shown is the average mean delay for the entire 24-hour simulation period versus the relative network load. Here the load is a scaling factor for the intercreation intervals in the traffic generators. A load of unity matches that of the measured PACS. A load of two indicates that the intercreation intervals are halved (ie, traffic doubles).

From the PACS database for the 4 months described above, the average daily image volume generated by CT_B was determined to be 382.62 ± 104.14 Mbytes. The CT_B traffic generator produced 375.34 Mbytes for a single 24-hour simulation period, agreeing well with the average derived from the PACS database. Because the traffic generators are stochastic in nature, another 24-hour simulation period would produce somewhat different results. Similar results were obtained for the other traffic generators. Clearly, a vast majority of the image traffic is generated by the CR devices and also by the laser film scanners. The cumulative

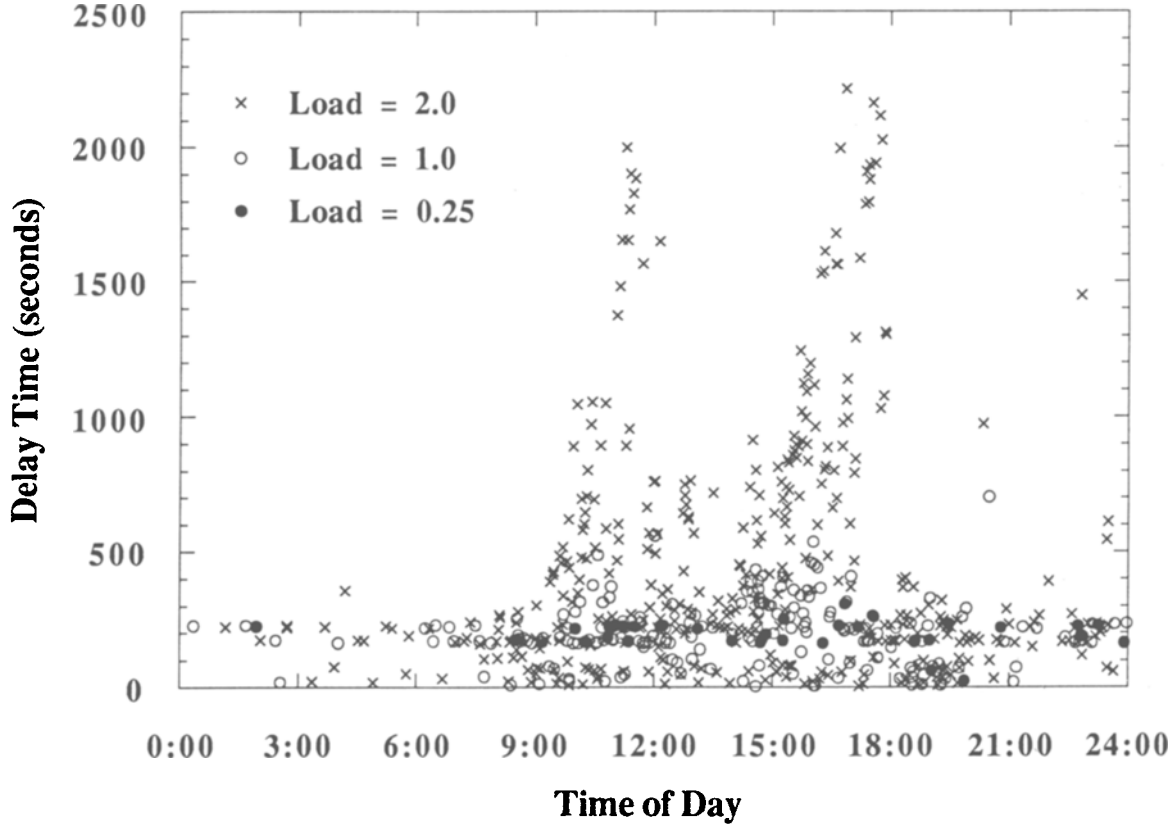

Fig 10. The graph given in Fig 9 is essentially "averaged" over the 24-hour period, thus it does not provide an idea how the delay time changes during the day. This graph shows the delay time for each file created during the 24-hour period versus the simulation clock. Data for relative network loads of 0.25, 1.0, and 2.0 are given.

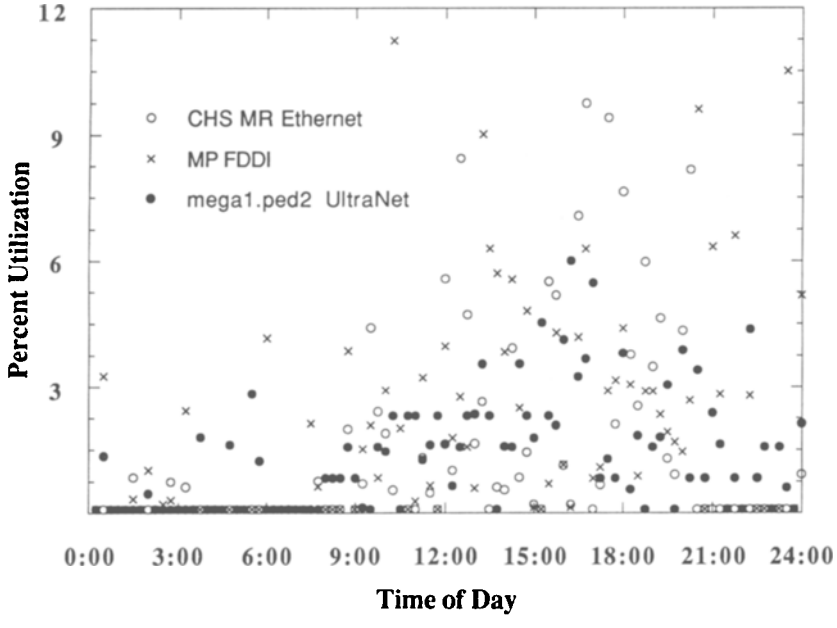

Fig 11. This graph shows how the three networks (Ethernet, FDDI, and UltraNet) are utilized for 15 minute intervals versus the simulation clock.

megabytes received by the image display station (over 1.2 Gbytes) indicates that large local disks are required for the proposed architecture.

Mean Delay Versus Load

A critical measurement in this simulation is the mean delay time. Delay time is measured as the difference in time for image transfer from the imaging modality until the image is staged in the image display workstation, ready for display. Figure 9 shows the mean delay of the entire system for the 24-hour period as a function of load. Here the load is a scaling factor for the intercreation intervals in the traffic generators. A load of unity matches that of the measured PACS. A load of two indicates that the intercreation intervals are halved, doubling the image traffic.

The mean delay for the modeled PACS is found to be 163 ± 100, 225 ± 59, and 360 ± 202 seconds for respective loads of 0.25, 1.0, and 2.0. This delay time is dependent on overall system load, the stochastic nature of the generated traffic patterns, as well as file length. The increase in the mean delay time greater than a relative load of 1.5 indicates the presence of a bottleneck in the system, which was determined to be the acquisition and reformatting computers. Mean delay is averaged over the entire 24-hour period and, although it gives a general estimate of the delay time, it does not incorporate the effect of varying system load throughout the 24-hour period.

Delay Time Versus Simulation Clock

Because a traffic generator's intercreation interval varies during the 24-hour simulation, the delay time also varies with the simulation clock. The graph given in Figure 9 is essentially "averaged" over the 24-hour period, thus it doesn't provide an idea how the delay time changes during the day. Figure 10 shows how the delay time varies with the simulation clock for a relative load of 0.25, 1.0, and 2.0. The delay time does not vary much over the simulation clock for a load of 0.25, however, for a load of 1.0, the delay time peaks at approximately 10 AM and 3 PM, but the delay time does not increase very much. During busy clinical hours (9 AM to 5 PM), the delay time lengthens substantially for a load of 2.0, especially at approximately 10 AM and 3 PM which are traditionally heavy utilization times for the CR and LS modalities. For a load of 2.0, the two large peaks at approximately 10 AM and 3 PM yield delay times as large as ten times that of the unity load mean delay. Extremely long (up to half-hour) delays are experienced by some files during these times for a load of 2.0 indicating a serious bottleneck in the system which was alluded to in the increase of the average delay time as load increased greater than a load of 1.5, as shown in Fig 9.

Utilization Versus Simulation Clock

Figure 11 shows how the various portions of the three networks (Ethernet, FDDI, and UltraNet) are used during the 24-hour period for a load of unity. An average use over the entire 24-hour period only tells a small fraction of the entire story. Therefore, the 24-hour period has been broken down into 15-minute intervals. It is the average use over these quarter-hour intervals that are plotted in Fig 11.

The Ethernet use is for the module connected to the MR_C traffic generator in the Center for Health Sciences (Fig 4, bottom left). The mean Ethernet use is 1.4% (SD, 2.4%) with a maximum of 9.7%. The FDDI use is for the FDDI module connected to the CT_A, CT_B, MR_A, and MR_B in the Medical Plaza (Fig 4, top left). The mean FDDI use is 1.9% (SD, 2.5%) with a maximum of 11.2%. The UltraNet use is for the UltraNet module connected to the pediatric outpatient display station module (Fig 4, top right). The mean UltraNet use is 1.3% (SD, 1.4%) with a maximum of 6.0%.

CONCLUSIONS

The modular nature and graphical user interface of the BONeS software package simplifies the mechanics of programming a computer simulation, allowing greater time for the more critical stages like simulation model and performance metric definition. This simulation required less than 2 central processing unit minutes to complete each 24-hour period on a SPARCstation 2 (Sun Microsystems; Mountain View, CA).

The BONeS simulation model (Fig 4) of the PACS image management network (Fig 1) was very accurate. The models for the imaging

modality traffic sources were validated with a high degree of accuracy (Fig 8) indicating that the mean file lengths and mean intercreation intervals derived using the PACS database were analyzed correctly.

This simulation model has allowed us to study "what if" questions, such as what happens to the delay time if the traffic loading doubles (Figs 9, 10). Additional questions under study include, eg, how many image acquisition and reformatting computers are required to keep the mean delay time down to 5 minutes, 10 minutes, or 30 minutes, to keep the system cost down to a specified level.

The BONeS simulation has been used to predict potential bottlenecks, ie, portions of the network that have an extremely high utilization. In this simulation, the reformatting process was determined to be the bottleneck causing the large increases in delay time for large loads (Figs 9, 10).

The networks have been modeled herein as queues with file length dependent service times. The network portions of the simulation are currently being modeled with fine-grained resolution. This involves, for example, modeling network packetizing, protocol functions, and contention. Although they are not expected to change the results significantly for the delay time, as observed PACS throughput rates were used, the results for network use and throughput will prove to be much more detailed and useful.

BONeS has been used to simulate the throughput of a proposed teleradiology system[29,30] and to model patient throughput in our radiology department. For the radiology department throughput model, bottlenecks are being explored in the imaging of patients and traffic patterns that would indicate the best allocation of radiology personnel and acquisition of imaging equipment.

REFERENCES

1. Jain R: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling. New York, NY, John Wiley & Sons, 1991, pp 393-411

2. Robertazzi TG: Computer Networks and Systems: Queuing Theory and Performance Evaluation. Berlin, Germany, Springer-Verlag; 1990, pp 1-2

3. McHaney R: Computer Simulation: A Practical Perspective. San Diego, CA, Academic, 1991, pp 15-31

4. Pidd M: Computer Modelling for Discrete Simulation. New York, NY, John Wiley & Sons, 1989, pp 23-54

5. Law AM, Kelton WD: Simulation Modeling and Analysis. New York, NY, McGraw-Hill, 1982, pp 4-9

6. Payne JA: Introduction to Simulation: Programming Techniques and Methods of Analysis. New York, NY, McGraw-Hill, 1982, pp 10-26

7. Russell EC: Building simulation models with SIMSCRIPT II.5 Los Angeles, CA, CACI Products Company, 1983, pp 2-1–2-22

8. Pooley RJ: An introduction to programming in SIMULA. Boston, MA, Blackwell Scientific, 1987, pp 1-5

9. Pritsker AB: Introduction to Simulation and SLAM II. New York, NY, Halstead, 1984, pp 62-64

10. Danthine AAS: Protocol representation with finite-state models. IEEE Trans Comm COM 28:632-643, 1980

11. Neuts M: Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach. Baltimore, MD, Johns Hopkins University, 1981

12. Garg K: An approach to performance specification of communication protocols using timed petri nets. IEEE Trans Software Engin SE 11:1216-1225, 1985

13. Sauer CH, MacNair EA, Kurose JF: Queueing network simulations of computer communications. IEEE J Selected Areas Comm SAC 2:203-220, 1984

14. Frost VS, LaRue WW, McKee AG, et al: A tool for local area network modeling and analysis. Simulation 55:283-298, 1990

15. Cope P: The case for network modeling. Network World 8:1-5, 1991

16. Shanmugan KS, Frost VS, LaRue W: A block-oriented network simulator (BONeS™). Simulation 58:83-94, 1992

17. Shanmugan KS, Titchener P, Newman W: Simulation-Based CAAD Tools for Communication and Signal Processing Systems. Proc IEEE Int Conf Comm, Boston, MA, June 11-14, 1989

18. Horii SC, Bidgood WD: Network and ACR-NEMA protocols. RadioGraphics 12:537-548, 1992

19. Bidgood WD, Horii SC: Introduction to the ACR-NEMA DICOM standard. RadioGraphics 12:345-355, 1992

20. Stewart BK, Taira RK: Database architecture and design for PACS, in Huang HK, Ratib O, Bakker AR, et al (eds): Picture Archiving and Communication Systems (PACS) in Medicine. Berlin, Springer-Verlag, 1990, pp 83-90

21. Taira RK, Stewart BK, Sinha U: PACS database architecture and design. Comput Med Imaging Graphics 15:171-169, 1991

22. Dixon WJ, Massey FJ: Introduction to Statistical Analysis (ed 4). New York, NY, McGraw-Hill, 1983, pp 372-377

23. Stewart BK: Three tiered network architecture for PACS clusters, in Huang HK, Ratib O, Bakker AR, et al (eds): Picture Archiving and Communication Systems (PACS) in Medicine. Berlin, Springer-Verlag, 1990, pp 113-118

24. Stewart BK, Honeyman JC, Dwyer SJ: Picture archiving and communication system networking: Three implemen-

tation strategies. Comput Med Imaging Graphics 15:161-169, 1991

25. Stewart BK: Network topologies, media and routing. Radiographics 12:549-566, 1992

26. Stewart BK, Lou SL, Wong WK, et al: An ultra-fast network for radiological image communication. AJR Am J Roentgenol 156:835-839, 1991

27. Stewart BK, Lou SL, Wong A, et al: Performance characteristics of an ultrafast network for PACS. Proc SPIE 1446:141-153, 1991

28. Stewart BK, Taira RK, Dwyer SJ, et al: Acquisition and analysis of throughput rates for an operational, department-wide PACS. Proc SPIE 1654:24-38, 1992

29. Stewart BK, Dwyer SJ: Teleradiology system analysis using a discrete event driven block oriented network simulator. Proc SPIE 1654:2-13, 1992

30. Stewart BK, Dwyer SJ: Discrete event driven block oriented network simulation applied to teleradiology system analysis. Invest Radiol 28:162-168, 1993