Journal of Digital Imaging

Predicting Clinical Image Delivery Time by Monitoring PACS Queue Behavior

Nelson E. King,[1] Jorge Documet,[2] and Brent Liu[2]

The expectation of rapid image retrieval from PACS users contributes to increased information technology (IT) infrastructure investments to increase performance as well as continuing demands upon PACS administrators to respond to "slow" system performance. The ability to provide predicted delivery times to a PACS user may curb user expectations for "fastest" response especially during peak hours. This, in turn, could result in a PACS infrastructure tailored to more realistic performance demands. A PACS with a stand-alone architecture under peak load typically holds study requests in a queue until the DICOM C-Move command can take place. We investigate the contents of a stand-alone architecture PACS RetrieveSend queue and identified parameters and behaviors that enable a more accurate prediction of delivery time. A prediction algorithm for studies delayed in a stand-alone PACS queue can be extendible to other potential bottlenecks such as long-term storage archives. Implications of a queue monitor in other PACS architectures are also discussed.

KEY WORDS: PACS system performance, image retrieval delivery time, response time, benchmark

INTRODUCTION

PACS designers minimize the delivery time of images to users at their workstations through a number of technologies including high-speed networks, fast servers, and prefetching algorithms. Researchers aid these designers by developing monitoring tools to identify bottlenecks (e.g., Nagy et al.)[1] between a PACS and workstations on a local network. Careful selection of hardware and use of monitoring tools prevent most problems with slow delivery during initial deployment. However, PACS usage inevitably grows over time resulting in slower delivery times each ensuing year.

Most solutions to slow delivery times have focused on increasing the ability to "supply" images such as faster hardware for storage, servers, networks, and workstations. However, a cost-effective PACS implementation will unlikely be able to accommodate "spikes" of maximum usage during peak hours. As a consequence, there will be limited periods of peak use where delivery times are slower than normal. Not knowing the expected delivery time may lead to unreasonable expectations. Sustaining reasonable image delivery times during peak usage necessitates managing the "demand" for images or investing in expensive hardware upgrades for a few short bursts of peak usage.

One technique for managing demand is to inform users about the expected delivery time of an image request. Computer usability guidelines recommend progress indicators to mitigate user expectations when tasks take more than a few seconds such as in downloading files or a system with slow response resulting from busy servers.[2,3]

[1]From the Olayan School of Business, American University of Beirut, Bliss Street, P.O. Box 11-0236, Riad El-Solh/Beirut, 1107 2020, Lebanon.

[2]From the Image Processing & Informatics Laboratory, Department of Radiology, University of Southern California, Marina del Rey, CA 90292, USA.

Correspondence to: Nelson E. King, Olayan School of Business, American University of Beirut, Bliss Street, P.O. Box 11-0236, Riad El-Solh/Beirut, 1107 2020, Lebanon; tel: +961-1-374374-3731; fax: +961-1-750214; e-mail: linking@acm.org

Online publication 12 April 2006
doi: 10.1007/s10278-006-0262-z

A queue monitor is a tool that informs users of predicted delivery time when a backlog of requests is developing. Users could then choose to take a short break or temporarily switch to other tasks rather than waiting. Although some current PACS can provide the administrator with the status of a series (e.g., pending, sending) or users an image counter (e.g., sending image 3 of 59), none—to our knowledge—can predict delivery time.

This article begins with a description of our laboratory test bed that allowed us to identify the PACS parameters to be used in a queue monitor prediction algorithm. The section Data summarizes the simulation data that illustrates the interaction among the parameters important to predicting delivery time. The results and implications to PACS designers and system implementers are discussed in Results section.

MATERIALS AND METHODS

Our goal was to identify the PACS parameters that play a part in predicting the delivery time of a study under conditions of queuing such as during peak usage in a clinical setting. These parameters would be used to develop a delivery time prediction algorithm by monitoring a PACS queue. We simulated a clinical environment in which the requests to retrieve clinical images from our laboratory's PACS Simulator resulted in requests being queued rather than sent immediately. The queuing behavior could then be examined under various conditions. The prediction algorithm would be derived from the performance data for this PACS, particularly with respect to the PACS queue. The study assumption was that saturating a PACS with delivery requests would result in an observable backlog of requests in the queue. A slow PACS can be saturated with much fewer requests for studies than a faster PACS. However, a faster PACS could eventually be saturated with more study requests and larger studies (e.g., hundreds of slices). Demonstrating a queue backlog with a slow PACS should therefore be generalizable to a fast PACS.

A laboratory test bed that modeled a clinical setting allowed us to analyze PACS parameters without disrupting an actual clinical operation. Actual clinical images were requested at 5- to 10-minute intervals to represent the typical workflow of a radiologist. Sometimes, multiple studies were requested at about the same time to reflect the condition when prior studies had to be retrieved for comparative purposes. Workstation Test Bed Description section describes the laboratory test bed and the workstations used. Data Acquisition summarizes the clinical data used in the study. Algorithm Description identifies the parameters to be used in the prediction algorithm.

Workstation Test Bed Description

A clinical setting was simulated in our IPI Laboratory. A PACS and five workstations running DICOM client software were placed on two different network segments. Three of the workstations were placed on the same 100 Mbps network segment as the PACS representing the conditions found in a reading room. Two other workstations were placed on a different 10 Mbps network segment representing workstations for clinicians or radiologists in a different part of the building.

The PACS Simulator in the IPI Laboratory was used for the PACS in the study. The software runs on an Ultra 2 Sun machine using SunOS 2.8. The database was Oracle 8i. The PACS Simulator is DICOM-compliant using the standard patient—study—series data model. There are four internal queues to which study requests are sent. Reports by Law and Zhou[4] and Zhou et al.[5] provide a more complete description of the PACS Simulator.

A PACS running on slower hardware was chosen so that fewer workstations could generate enough requests to saturate the PACS and cause some study requests to be delayed. In addition, placing two workstations on a slower network segment was expected to force more requests to be queued. The configuration of the five Windows-based computers that acted as reading workstations is shown in Table 1.

Data Acquisition

The data collection objective was to saturate the PACS server with DICOM send requests (i.e., C-Move) so that some study requests would be queued. The sequence of requests and time between requests should be representative of a clinical setting. This precludes selecting a large number of exams via a wildcard search and requesting all the resultant studies to be sent all at once. Such an approach would greatly distort the results particularly with the queue algorithm of our PACS Simulator. Studies transferred under a single query are all sent to the same queue within the PACS Simulator even though four queues are available. A backlog in a single queue is created because processing is performed sequentially. Thus it was necessary to manually query and retrieve every study by name from a workstation.

Table 1. Workstation specifications in queue monitor testbed

Workstation	DICOM client	OS	CPU	Network	Database
Client1	Conquest 1.4.7	XPP	P4, 2.8 GHz	Same as PACS (100 Mbps)	Built-in DbaseIII
Client2	Conquest 1.4.7	XPP	P3, 0.7 GHz	Same as PACS (100 Mbps)	Built-in DbaseIII
Client3	Cedera I-View 5	W2K	P4, 1.3 GHz	Same as PACS (100 Mbps)	Vendor-provided
Client4	Conquest 1.4.7	W2K	P4, 2.8 GHz	10 Mbps subnet via firewall	Built-in DbaseIII
Client5	Conquest 1.4.7	W2K	P2, 0.4 GHz	10 Mbps subnet via firewall	Built-in DbaseIII

The data collection procedure was to request a study from each of the five workstations until all of the anonymized data had been transferred from the PACS server to a workstation. The testing sequence began by initiating a query from the first workstation (e.g., Client1) to PACS for locating a specific study. Once the results of the search are returned from PACS, a transfer was requested from the PACS to this workstation. The second workstation then queried a different exam based on study number. Once again the transfer was requested immediately after the query result was returned by the PACS. The same procedure was done for workstations 3, 4, and 5. The researcher would then return to workstation 1 and query another exam. The transfer was initiated and the process repeated on workstation 2. Sometimes additional exams were requested on a client to simulate the retrieving of prior studies for comparison. Table 2 tabulates the breakdown of 131 studies that were requested over a 40-minute period. Nearly 2 GB of data was transferred and almost 4,500 individual DICOM images were sent. There were studies from modalities of computed radiography (CR), computed tomography (CT), magnetic resonance image (MRI), a single ultrasound (US) study, and a CR study containing two highly compressed files.

After all of the requested studies were processed on the five workstations, the queue transaction log of the PACS Simulator was accessed using an SQL client and an ODBC connection. The data preserved in the queue log included DICOM requestor and recipient, assigned queue, time of retrieval request, start time of retrieval, finish time of retrieval, and specific study identifier.

Algorithm Description

The prediction algorithm takes into consideration site-specific conditions. These include network segment, client computer capability, DICOM application software, load on the PACS, and specific file parameters of a clinical study. All of these factors affect the transfer rate of images to a workstation. *Network segment* affects the speed of transfer depending upon bandwidth. In addition, the file may pass through any combination of firewall, switch, or hub. *Computer capability* determines the speed at which our DICOM client software could receive and store images locally. The DICOM headers had to be analyzed and the metadata stored as a record on a local database. *Application software* affects transfer speed by choice of database and processing algorithms. *Load on the PACS* reflects the capability of the PACS hardware and software to send multiple studies at the same time. There may be limitations on throughput due to network card, buffer or thread implementation, server memory, and processing power. *Clinical study specifics* include modality, number of images or slices, and compression. CR studies have a few large images, which means that transfer rates are dominated by the number of bytes to be transferred. In contrast, MR images are small (256 × 256), but in tens if not hundreds of images, which means a greater percentage of time is spent on header processing and DICOM file transmission overhead.

DATA

A log from the PACS Server was generated from the simulation requesting 131 studies from five workstations over a 40-minute interval beginning at 46,000 seconds (12:47 P.M.) to 48,500 seconds (1:28 P.M.). The analysis of this log is presented in this section.

A 400-second interval from the total 2,500-second simulation is shown in Figure 1, graphically illustrating the sequence of studies transferred for the five clients. The horizontal axis is expressed in seconds for that day (e.g., interval starts at 1:07:30 P.M.). The time to transfer a study consists of a delay in the start of transmission (i.e., sitting in queue) and the actual send (transmitting) time. The chronology of the first 13 studies in this figure is explained, beginning from the bottom. Client1 and Client2 already have a study transfer in progress at time 47250, which is why no delay time is evident. Client1 then makes three additional requests within a 16-second period beginning at time 47251. The next request by Client1 takes place at 47501 seconds. Client2 makes a single request at 47283 seconds. Six minutes later, Client2 requests a set of six studies. The 14th to 26th study represents the studies requested by Client3, Client4, and Client5.

The remainder of this section consists of descriptive statistics from this simulation. Performance characteristics of each workstation are contained in

Table 2. Distribution of studies by workstation destination

				Modality				
Workstation	Total Mbytes	Total images	Studies	CR	CT	MR	US	CR[a]
Client1	372	920	25	3	13	7	1	1
Client2	355	918	25	3	13	7	1	0
Client3	455	892	29	7	14	6	1	1
Client4	390	931	25	5	11	8	1	1
Client5	444	803	26	7	11	6	1	2
Total	2,012	4,464	131	25	62	34	5	5

[a] Compressed.

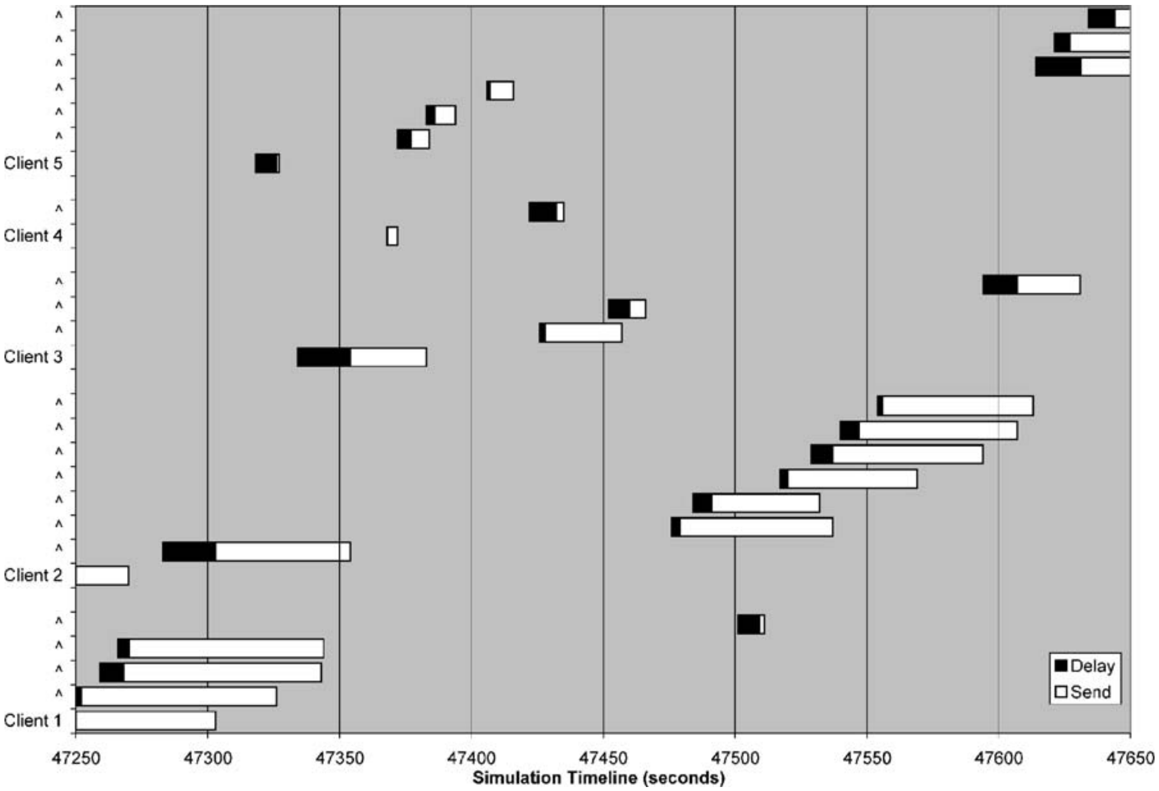

Fig 1. Sequence of study transfers over 400 seconds.

Workstation Performance section. Workstation and the Impact of Modality describes the impact of modality upon workstation performance. The reader should note that 13 of the 26 delays in the interval shown in Figure 1 resulted from a queue backlog that is discussed in Backlog in Queues.

Workstation Performance

The differences in transfer rates to each workstation reflect a number of parameters important for a delivery time prediction algorithm. These include the computer components of the workstation, network segment, and application software. Table 3 provides average performance for each client showing relative differences between the workstations without regard to modality (see Workstation and the Impact of Modality). The standard deviation is shown in brackets for each value.

The average time for processing a study in the first column ranges from 10.2 seconds for Client4 to 53.6 seconds for Client1. This is the time from when the PACS received a request to the time that PACS records in the log that the study has been sent. The average transfer rate is measured in megabytes per second (MBps) as shown in the second column, where the total size in megabytes of the study is divided by the time from when the PACS begins transmitting the images to the workstation until the study is completed. Client1 and Client2 have very slow transfer rates because of their location on a 10-Mbps (1.25 Mbps) network segment. The third set of measures is the average start delay in seconds as shown in column 3. Start delay (“delay”) marks the time when PACS receives a request up to the time the transmission of images has begun. Our PACS Simulator uses four queues. There were 22 studies that became backlogged in a queue during the 40-minute simulation. Many of these 22 studies had lengthy delay times simply because they had to wait for the previous study to finish. The average delay in seconds excluding these 22 studies is found in column 4.

Workstation and the Impact of Modality

Modality determines the file characteristics of a study especially image size (MB) and number of

Table 3. Transfer rates by workstation destination (standard deviations in brackets)

Workstation	Average time for study (s)	Average transfer (Mbytes/sec)	Average start delay (s)	Average start delay without backlog (s)
Client1	53.6 [22.5]	0.34 [0.15]	7.0 [3.5]	7.0 [3.4]
Client2	48.1 [12.6]	0.38 [0.17]	7.1 [5.9]	5.2 [2.9]
Client3	23.5 [12.3]	2.01 [2.26]	7.8 [6.7]	5.4 [3.4]
Client4	10.2 [5.7]	4.21 [1.94]	6.5 [5.5]	5.3 [3.4]
Client5	17.0 [8.2]	1.72 [1.16]	6.2 [3.8]	5.9 [3.3]
All	29.8 [21.5]	1.77 [2.01]	6.9 [5.2]	5.7 [3.3]

images. A CR study has large images (e.g., 8 MB) but only a few images. An MR study has small images but many slices (e.g., 50–60 in this study). Because our workstations functioned as local DICOM servers, the images were both transferred to the hard disk of the workstation and catalogued. This meant the headers on each DICOM image had to be processed so that storage could take place locally. Hence, more workstation resources are spent processing an MR study than transferring the image. This is why the transfer rates for CR are typically higher than CT and MR. This is shown in the average transfer rate column of Table 4. Standard deviation is shown in brackets. The anomalous result of Client1 for CR reflects a relatively rare situation where all three CR studies in the simulation for Client1 were retrieved simultaneously. Backlog in Queues explains how simultaneous transfers decreased throughput by nearly one-half.

Backlog in Queues

A queue backlog occurs when there are more requested studies than queues to process the requests. Figure 2 shows the concept of a queue backlog taken from the study data from time 47250 to 47500. Client1 workstation requested a study at 47247. This request is assigned to Queue 2 finishing at 47326. The next study request assigned to Queue 2 comes from Client5 workstation. The request occurs at 47318 but cannot start sending until 47327 because each queue processes requests sequentially. Similarly, Client2 workstation requests a study at 47283, which is placed in queue 3 and finishes at 47354. The study requested by Client3 workstation at 47334 must wait in queue 3 until 47354 to begin sending. This figure also shows that our PACS Simulator distributes studies evenly across the queues. For example, three requests from Client1 workstation made within 20 seconds of each other (47247 to 47267) are sent to queues 1, 2, and 4.

Exclusion of delays caused by the 22 incidents (29% of requested studies) of queue overlap reduces the average delay time (averaged for all the studies) from 6.9 to 5.7 seconds (e.g., last row in last two columns on Table 3). All delay times exceeding 12 seconds (10 incidents) were attributed to queue overlap. The standard deviation (average for all the studies) drops from 5.2 to 3.3 seconds.

RESULTS

The queue backlog condition that was simulated in this research allows several types of findings to be discussed. Several critical assumptions and

Table 4. Transfer rates by modality and destination (standard deviations in brackets)

	Transfer rate (Mbytes/sec)		
Workstation	CR	CT	MR
Client1	0.3 [0.03]	0.4 [0.1]	0.3 [0.2]
Client2	0.6 [0.2]	0.4 [0.1]	0.2 [0.1]
Client3	5.1 [2.5]	1.4 [0.8]	0.3 [0.1]
Client4	6.4 [1.2]	5.0 [0.6]	2.2 [1.0]
Client5	3.3 [0.6]	1.7 [0.4]	1.7 [0.4]
Average	3.8 [2.6]	1.7 [1.7]	0.8 [1.0]

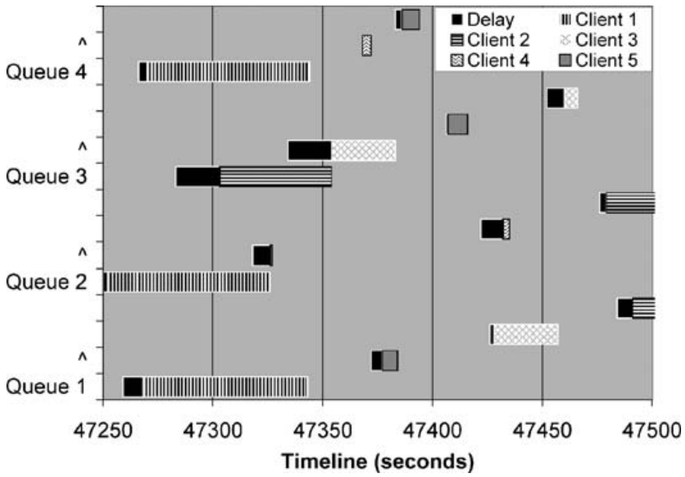

Fig 2. Delays caused by overlapped studies in a queue.

parameters of importance for at least our laboratory configuration are discussed in Important Parameters. The framework of a prediction algorithm is described in Prediction Algorithm. Implications for a Clinical System discusses the implications of these findings on clinical systems. We discuss future research plans based on our findings in Future Research.

Important Parameters

Our simulation results reflect an important data collection assumption—that each study request had to be queried individually. A unique study number had to be manually typed in to query and then retrieve the study. This protocol was adopted after observing that a query command retrieving multiple studies (e.g., all studies of patient *xyz* or studies containing string 1234) placed all of the selected studies in the same queue of our PACS Simulator. This places multiple study requests in a single queue forcing a queue backlog to develop. The transmission rate to workstations would then be limited by the sequential processing of the queued requests regardless of the state of the three remaining queues. It should be noted that our PACS can process requests made from multiple workstations at the same time, just not multiple studies requested at the same time from the same workstation.

Some might argue that this protocol assumption was somewhat artificial. After all, a radiologist might wish to retrieve all studies for a particular patient before starting to read an examination. Or a PACS might be configured with the policy to prefetch all studies for a patient. Our point is that the implementation approach chosen by our PACS developers for requesting multiple studies at the same time will increase the duration of transfer. All of the studies made in a single request ended up in the same queue. A more effective implementation is to assign each study requested to its own queue or thread.

The bottleneck in our laboratory configuration was a slow 10 Mbps network segment where Client1 and Client2 resided. The transfer rate is slow to these two workstations, which lengthened the time a study resides in the PACS while being processed. Because our protocol required studies to be requested individually, more than one study could be transferred at one time to a workstation. For these two workstations, other studies were being simultaneously processed 88% and 76% of the time, respectively. In contrast, the other three workstations on the fast network (e.g., 100 Mbps) received the studies very quickly, which reduced the incidents of a study being transferred at the same time to between 19% and 28% of the time.

The key parameters encountered in this simulation that can be used in predicting delivery time consist of two types. The first type of parameters are related to physical attributes such as network configuration, image type (e.g., modality), and workstation hardware. Choice of network segment has a large impact for two reasons. First, the studies may need to pass through additional network devices such as firewall or switch. Second, network contention further slows transfer rates when studies are being simultaneously transmitted. Client1 and Client2 on a network segment passing through both a firewall and 10 Mbps switch were markedly slower than Client3, Client4, and Client5 sitting on the same segment as PACS. Transfer rates were also markedly different by modality type (Table 4) for all workstations. The configuration of workstation hardware is a parameter when the DICOM client must process the header and store the files locally. For example, Client4 (2.8 GHz CPU) transfers studies several times faster than Client5 (0.7 GHz CPU).

The second parameter type affects the delay in starting transmission of a study. The presence of a queue backlog is one such parameter when one or more studies are waiting for the queue to finish transmitting a study. The transfer rate between the PACS and the workstation can be used to determine how long the study in the queue will take to complete. We found by examining each study request (total of 131) that delay times exceeding 12 seconds were always caused by a queue backlog. Another parameter is the number of simultaneous studies being transmitted because each study being processed consumes CPU resources. The number of DICOM query requests will also consume CPU resources, consequently slowing the processing of a study request. The varying consumption of CPU resources is the likely source of the 3.3 seconds standard deviation in average delay time (for all studies) of 5.7 seconds (i.e., 2.4–9.0 seconds for 1 standard deviation) as shown in bottom row of last column in Table 3.

Prediction Algorithm

The prediction algorithm for a typical study request in which the PACS is not overloaded is quite simple. The algorithm using standard programming style for variable names is:

predictFinishTime = *predictSendStart* (= delay time) + sending time based on transfer rate of the workstation on its network segment given the particular study specifics (*transferRateClient*) and the load on a PACS sent to a particular workstation configuration.

For example, Client5 requests a CR study consisting of two 8-Mbyte images. A value of 5.7 ± 3.3 seconds will be used for the delay. Table 4 indicates a transfer rate of 3.3 Mbytes/second under average conditions. In the case of Client5 during this simulation, about 20% of the studies were simultaneously done with others. For the condition of a lightly loaded PACS Simulator, the predicted finish time is 5.7 + (16/3.3) ~ 10 seconds. However, what happens if PACS is busy and the request from Client1 arrived before the request from Client5? The request is for a 200-slice MR (about 27 Mbytes), which means a download at 0.3 Mbytes/seconds. Thus, after 90 seconds (27/0.3), the backlogged request can begin. The predicted delivery time would then be 95 seconds.

Implications for a Clinical System

A properly designed clinical system means minimal periods of peak usage conditions. Hence, a queue monitor will not be used frequently. However, there are numerous possibilities in which unplanned heavy usage could make use of a queue monitor valuable. For example, digital mammography generates very large files that take many seconds to transfer to or from a PACS. A queue backlog could be generated during this transfer period as one of the queues is dedicated to this large study. Usage often peaks during start of a shift or after lunch. Thus, a contingency plan is necessary—buy more hardware, fine-tune the existing system, and notify users of peak usage. A queue monitor helps operators to understand the behavior of the system—such as our experience of workstation performance being degraded not only by the network configuration but the load on PACS.

PACS architectures described by Huang[6] are increasingly client-server or web-based rather than stand-alone. Although the transport mechanisms and protocol of the image files being transferred may change with a client-server or web-based architecture, the underlying basis of queue monitoring does not change. There must be a queuing mechanism implanted in any PACS server to accommodate a backlog of requests. Status indicators on PACS clients that already provide feedback in the form of a showing that 1, 2, ... of *N* images have been received is a first step. Adding predicted delivery time is the next step. A client-server PACS architecture may actually facilitate the use of a queue monitor. The connection between client and server could be used to measure actual transfer rates and provide predictions to the users. The PACS in a client-server architecture has more information about the state of clients. These vendors can more easily incorporate software code into the client that gives users insight on study transfer times and the performance of the PACS.

Testing a stand-alone PACS with peak load conditions as done is this research will be difficult in a clinical environment. The speed of current PACS hardware makes it difficult to overload the PACS with requests so that a backlog in the queues is created under ordinary conditions. Access to many workstations would be required with near-simultaneous requests for studies being made. One approach during acceptance testing to simulate a peak load would be to transfer very large studies with thousands of images from as many clients as practical. The lengthy transfer time of a very large study fills one queue slot so that other requests are sent to other queues. Fewer workstations and the personnel to operate them would be needed to create a peak load.

When selecting a vendor, questions should be asked about the means in which connections are established between the client or web application and the PACS server. Our PACS placed all the requests for multiple studies (e.g., using wildcard search) in the same queue for transmission as only a single DICOM connection was established. We have concerns that increasing interest among clinicians for web access means most of the requests are coming from browsers rather than reading workstations. This means a PACS architecture should be examined to determine how web and client connections are established with the PACS. For example, a typical implementation with web clients (e.g., DataServer[7]) allows a maximum of ten simultaneous connections with further requests queued or even rejected if the queue is full.

The implementation of a queue monitor requires that the PACS software store sufficient information to assess the state of the queues. This information does not need to be in one table of the database. However, the information about these queues must be queried in real-time (read-only) from a queue monitor application. Unfortunately, a prediction algorithm in the form of an SQL query would have to be developed for every PACS because there are vast differences in implementation and database structure. A read-only query minimizes the load on the PACS server. Ideally, PACS developers would write all of the critical queue information in a single table in a standard format. By doing so, a read-only table copy would only be necessary rather than the computation-intensive joins needed in an SQL query.

Our research focused on queue backlogs for PACS and the transmission time of images. Yet, the longest delays are probably not in the transfer of image files. Once files have begun to be transferred from PACS, the prediction of delivery time is straightforward. A greater benefit comes from monitoring the queues of long-term storage devices. For example, a radiologist may wish to retrieve a “prior” study several years old that has been automatically migrated to a tape silo. Prefetching is of course possible, but storage policy will inevitably limit a prefetch to more recent studies. There is a delay of tens of seconds (e.g., robot selects tape then streams tape to desired location) before the images can be transferred. If there are many requests to the tape silo, then the delay grows quickly as more requests are queued. Having this delivery time information, the radiologist can exercise other options besides waiting. For example, prediction of a long delay can be sent to the PACS. The PACS can in turn update the Radiology Information System (RIS), which could suggest to the radiologist that this patient be skipped for the moment.

Future Research

Our future research will be directed toward building a web application so that the prediction algorithms can be accessible from anywhere in the clinical setting. Other PACS queue implementations and PACS architectures will also be studied so that a generalized prediction algorithm can be written based upon different implementation approaches. A queue monitor architecture can then be proposed to a standards body. Other potential queue bottlenecks such as long-term storage or tape archives will also be incorporated into the algorithm.

Although the queue monitor manages user expectations by predicting delivery times, the prediction algorithm can also serve in a load balancing capacity as a back-end tool in the PACS infrastructure. Prefetching of studies likely to be needed that day can be retrieved in advance during slow periods especially off slower media such as tape.

The prediction algorithm could also be used in distributed storage networks (e.g., data storage grid[8]) to predict which storage resource is likely to provide the quickest delivery times. When images are distributed across a wide area network (WAN), delivery time depends on all the parameters considered in a local PACS architecture plus the actual speed of the connection to the remote site.

Because this study was conducted on a PACS that utilizes older hardware, future testing on faster PACS hardware will require a reading workstation simulator. The simulator must provide for multiple DICOM clients residing on different network segments that request studies individually over a time interval consistent with radiologist reading habits. Although some load on the PACS can be created with larger studies (hundreds instead of tens of slices), simulating peak load conditions will require the use of multiple DICOM clients. The workstation simulator will also be useful in establishing benchmarks for a particular PACS. We expect each PACS to have vastly different performance behavior under peak load conditions because of the means in which simultaneous studies are processed.

Our prediction algorithm considered average transfer rates. In reality, there will be a large number of workstations whose configuration will be known including hardware components, software, and network location. A self-learning algorithm seems ideally suited for a queue monitor application. There are many known variables about each workstation and the type of study being sent. Each of these sets could be associated with measurable PACS parameters such as processor and memory utilization as well as queue status (e.g., presence of queue backlog). The transfer rates could be periodically updated using all of the previous studies as a training set.

CONCLUSIONS

A queue backlog condition that results from too many requests for clinical studies from a PACS has implications beyond slow delivery of clinical images. We demonstrated the feasibility of a PACS Queue Monitor algorithm so that performance of our stand-alone PACS could be monitored under peak loads. Monitoring the queue of our PACS allowed us to predict delivery time potentially offering a means to “manage” user expectations. We found that fast performance from a PACS requires more than just fast networks and fast workstation hardware. The queuing implementation and the connection established between a PACS and a workstation is also a factor. Our prediction algorithm can predict approximate delivery times, but further understanding of PACS behavior under peak load conditions is necessary to refine the algorithm. In particular, monitoring the queues of relatively slow storage devices such as tape is necessary to accurately predict delivery time.

ACKNOWLEDGMENTS

The corresponding author thanks his colleagues at the Image Processing and Informatics Laboratory for stimulating his interest in medical imaging informatics. An earlier version of this article was presented to the 2005 SPIE Medical Imaging Conference. This work was partially supported by NIH Grant No. R01-LM06270 from the National Library of Medicine.

REFERENCES

1. Nagy PG, Daly M, Warnock M, et. al: PACSPulse: A web-based DICOM network traffic monitor and analysis tool. Radiographics 23(3):795, 2003

2. Nielsen J: Usability Engineering. (Morgan Kaufmann, San Francisco, 1994)

3. Galletta DF, Henry R, McCoy S, et. al: Web site delays: How tolerant are users?. J Assoc Inf Syst 5(1):1, 2004

4. Law MYY, Zhou Z: New direction in PACS education and training. Comput Med Imaging Graph 27:147, 2003

5. Zhou Z, Huang HK, Liu BJ, et. al: A RIS/PACS simulator with web-based image distribution and display system for education. SPIE Med Imaging 5:372, 2004

6. Huang HK: PACS and Imaging Informatics: Basic Principles and Applications, 2nd edition. Wiley-Liss, 2004

7. Bui AAT, Weinger GS, Barretta SJ, et al: DataServer: An XML gateway for medical research applications. In: Techniques in Bioinformatics and Medical Informatics, vol. 980. 2002, p 236

8. Liu BJ, Zhou MZ, Documet J: Utilizing data grid architecture for the backup and recovery of clinical image data. J Comput Med Imaging Graphics 29(2–3):95, 2005