A Study of Communication Delays for Web Transactions *

Bharat Bhargava
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907, USA

{bb}@cs.purdue.edu

Fax: +1-317-494-0739

Abstract

The performance of transactions originated on the world wide web depends on many factors. One of the major factor is communication delays involved in these web transactions. Mechanisms have been developed for studying the performance of the distributed transaction processing on the Internet, without *actually* having to move the database sites to remote Internet hosts. Experimental studies have been conducted to analyze and understand the behaviour of web based transactions and to measure the communication dalays involved in them. The developed mechanisms have been used to perform a series of experiments between Purdue and more than twenty Internet hosts around the world. Experiments were also conducted to measure the communication delays for different kinds of *Web Transactions* which includes web page downloading, digital library transactions and e-commerce transactions.This paper presents the experimental results for a few typical cases. It concludes by suggesting some directions for decreasing the communication latency.

Keywords: distributed databases, performance, Internet, communication, Web, Web Transactions.

1 Introduction

From its origin in 1991 as an organization-wide collaborative environment at CERN for sharing research documents in nuclear physics, the Web has grown to encompass diverse information resources: personal home pages, online digital libraries, virtual museums, product and service catalogs, government info for public dissemination, research publications, and Gopher, FTP, Usenet news and mail servers. The Web is increansingly being used in all aspects of society. For example consumers use search engines to locate and buy goods, or to research many decisions (such as choosing a holiday destination, medical treatment or election vote). The Internet and World Wide Web (WWW) have made electronic commerce a reality. According to an estimate the WWW presently contains more than 800 millions pages, encompassing about 6 terabytes of text data on

*This research is in part supported by the National Science Foundation CCR-0001712, CCR-001788, CERIAS and IBM SUR grant.

about 3 million servers. As the number of web users is increasing rapidly, there is a need to build fast and efficient systems for handling users requests on the web.

Internet users often experience performance variations among different kinds of web queries or *Web Transactions.* There are many factors which affect the performance of Web transactions. The main factors are communication delay, server processing capability and I/O time. The last two bottlenecks can be overcome by having multiprocessor systems and caching facility. The performance of HTTP, protocol used to access web page, is discussed in [20,24]. This paper does not focus on the HTTP protocol and mainly focuses on communication delays involved in Web Transaction processing. Communication delays depend upon a number of factors: network bandwidth, number of hops, network traffic, reliability of links, and hosts and network protocols. Though with the advent of new network technologies the bandwidth available is much larger than in past, the study reveals that the communication delay constitutes a major part of the response time for Web Transactions. This stresses the need to design efficient, reliable and scalable communication facility to provide improved performance over the Web. A comprehensive study and analysis of communication dalays for various kinds of web transactions will be of great help in designing and implementing faster web transaction processing systems.

Experiments have been conducted to study the communication performance for various kinds of Web transactions. These transactions are categorized into data-intensive and computation-intensive for the purpose of the studies. The correlation between communication delays and the size of the data has been studied, in addition to the correlation between communication and time of day and the communication delay to various sites in LAN and on the WWW. This paper presents the experimental results for texts and image data and for web page downloading. A novel technique called emulation via Wance tool is used to realize these experiments over the geographically dispersed sites. The study is concluded with suggestions to improve the communication delays involved.

2 Communication in the Internet

To study the performance of Web Transaction Processing (WTP) in WAN environment, the need for understanding the communication latencies and packet losses in an internetwork is essential. We have conducted the experiments to study the performance of message delivery in the Internet.

Based on our understanding of the Internet, we identified three factors that are important in assessing the performance of message delivery: the physical connection, the size of message, and the cross-traffic. The physical connection between two sites includes the distance, the type of links, and the number of hops (gateways). To establish the connection, we studied the performance across different sites on the Internet. We examined the effect of the size of the message transmitted to its transmission time. The delays and losses due to cross-traffic were quantified. A meaningful relationship with the communication performance could not be established due to uncontrolled and dynamically varying usage of Internet by various foreign agents. There is no easy way to determine the cross-traffic on a particular link. One way to circumvent the problem is to determine the traffic pattern at large, for the Internet. We suspected that it would be a function of the working hours, as the Internet usage determines the traffic pattern. To verify this, we examined how the time of the day and day of the week affects the message delivery performance in the Internet.

To summarize, we have conducted measurements in three dimensions: the *time* dimension by periodically repeating the experiments, the *site* dimension by repeating experiments with different sites, and the *size* dimension by varying the message sizes. We are interested in two performance measures: the round-trip time of a message and the message loss rate. In our model, round trip time is the time for a site to send a request message to another site and receive a reply message back. Message is said to be lost when the transport service of the Internet fails to deliver the message in time.

Our experiments involved over 2000 sites and 500 networks in the United States. We probed the Internet with ICMP and UDP messages periodically and collected the data [8]. Based on these measurements, we summarize as follows:

- We observed that there is a large variation in parameters such as communication delay and message loss. The variations exist in two dimensions: along the time axis and across the networks.
- We observed that the time of day has strong influence on the message delivery. The message loss rate is much higher in the noon working hours, and much lower in the early mornings. The round-trip time for a message, on the other hand, does not have a strong correlation with the time of the day, except for the hourly peeks. We believe that this is caused by the hourly jobs scheduled to run on gateways.
- We observed that the message delivery has an unbalanced performance across the wide area networks, although most of the hosts reported within 400ms round-trip. The *"clustering"* effect in the Internet is also observed. The communication between a site and many different sites on another local network has similar performance, which can be represented by any host on that network. Therefore, the latency between two networks can be used to estimate the communication delay between two hosts in these two networks.
- Finally, we observed that for small messages that can fit in an IP data gram without fragmentation, there is an approximate linear correlation between the transit time and the size of a message. However, the message loss is not affected by the size.

2.1 Impact on Web Transaction Processing

The communication performance over the Internet will have a significant impact on the performance of web transactions. The time to deliver a message over the Internet is a number of magnitude longer than in a LAN. While it takes only a few milliseconds to deliver a message in a LAN [2,7], on the Internet it is several hundreds of milliseconds to send a message across the continent [11]. This means that a transaction stays longer in the system, implying the larger lock holding time for data items to be updated. This leads to increased contention in the database, affecting the throughput adversely due to concurrency control.

The difficult problem of finding a "good" timeout value in a LAN environment is further aggravated in the Internet. A timeout is required by a Web Transaction Processing system to trigger special treatment for the transactions that are unable to complete in time. In a LAN environment, its value is usually determined as a constant multiplied by the number of read/write operations in a transaction. This flat timeout is not adequate in the Internet because both the physical distance (number of hops) and the location of the host affects the message delays and the loss rates. Furthermore, with the advances in the CPU and I/O technology, it can be discerned that more time will be spent by the transaction waiting in the message queues than performing actual computations. Thus, the timeout value , may be dependent on the number and the destinations of the remote messages.

Autonomous control over LAN allows the modification of the communication software to provide physical multicasting, light weight protocols, etc. [7]. Unless dedicated links or special networks are adopted, not much can be altered in the shared public WAN such as the Internet. The performance of message delivery is determined by traffic and various other factors beyond the designer's control. Therefore, the focus of improving the Web Transcation processing performance must shift towards reducing the number of messages exchanged.

Web transaction processing systems seldom adopt a transport service that guarantees reliable delivery because of the associated overheads [7,19]. This approach works fine because of the high reliability in the LAN environment. Message losses as high as 30% [8] can be observed in the Internet. This will lead to an increased transaction aborts resulting in a degraded system performance.

The Web Transaction Processing algorithms have to adapt to large variations in message latencies and losses in the Internet. For example, the time dependent and dynamic site-to-site performance data, and not just static data specified as a function of geographic location of the site, must be available to the replication control and surveillance control protocols [6]. However, the detailed cost matrix to reflect the network dynamics grows as $O(n^2)$ and will become unmanageable, for large number of sites . This problem can be tackled by taking advantage of the `clustering` effect in an internetwork, i.e. the latency between two networks can be used to estimate the communication delay between two hosts in these networks respectively.

2.2 Web Transactions : Categories

Web transactions are categorized into three main categories for the purpose of this study. Web transactions which involve communication of huge chunk of data. These are classified as *Data Intensive Web Transactions.* Most of the Digital Library transactions fall into this category. Web page downloading is another example of data intensive transactions.

There are transactions that involve higher need for server processing time. These are classified as *Computation Intensive Web Transactions.* Various e-commerce transactions are of this category [4]. This study focuses mainly on communication delays involved because even for e-commerce transactions the response time is dominated by the time to transfer the data over the Internet.

The third category consists of transactions which are both data intensive as well as computation intensive transfer of image and video data with compression and decompression[1,5].

2.3 Digital Library Transactions

Digital libraries provide a convenient means to access information over the Web. They provide on-line access to a vast amount of distributed text and multimedia information sources in an integrated manner. Digital libraries encompass the technology of storing and accessing data, processing, retrieval, compilation and display of data, data mining of large information repositories such as video, audio libraries, management and effective use of multimedia databases, intelligent retrieval, user interfaces and networking. Digital library data includes texts, figures, photographs, sound, video, films, slides etc.

In case of digital libraries, users, data and information processing are distributed over various sites in the Internet [1]. The size of the data objects involved in a digital library transaction over the web is enormous. The current network technology does not provide the bandwidth required to transmit gigabytes of digital library objects over the Internet. The low bandwidth results in large response times when digital library data is retrieved in an internetwork.

Digital libraries deal with a variety of problems that come into play due to scaling:

- **Size of Data:** Digital library data objects can be very large - a compressed video file can be easily 500 Mb. NASA image files of 1000K are not uncommon. Huge encyclopedia of text can be thousands of kilobytes. Retrieving these large data objects in a global distributed environment with the limited bandwidth available leads to an unacceptable response time in user interactions.

- **Number of Data Objects:** There are billions of large data objects. A NASA image database would contain millions of images. A database associated with a e-commerce would contain hundreds of thousands of product image clippings. books and journals.

- **Number of Sites:** The number of locations of information repositories available is increasing everyday. This can be observed by the 300% increase in World Wide Web servers in the past year [1]. A transaction requires access to many sites containing data than a traditional database where only specialized users access data. A site could be where a user is located or database is stored.

- **Number of Users:** The Global Information Infrastructure visualizes every home with a computer with easy access to the information highway. The number of new Internet users will increase with home computers.

Experiments have been performed to study the impact of communication in providing web access to distributed digital library over the Internet [5]. The performance of transmitting large, multimedia data objects across the Internet has been studied, with an objective to answering the following questions:

- What is the communication delay to different sites in the world?

- What is the correlation between the communication and the size of the message?

- What times of day are better for accessing information? What is the implication of different time zones on global communication?

Answers to these questions are important parameters in the development of web transaction processing systems. For example, a high communication delay indicates that some technique like retrieving a smaller version of the object, or a summary of the object should be used. Another example is that network traffic can determine the cost charged to a user [4]. Thus cost of access at night can be lower when compared to the cost in the day as is the practice followed by telephone companies.

3 Experiments on Web Page Downloading: Non-Secured vs Secured

A frequently issued transaction over the Internet involves web page downloading. The users experience poor performance while downloading a web page. To investigate the causes of performance problems, The steps involved in downloading a Web page and the sources of bottleneck have been examined [12]. Steps involved in downloading a Web page are DNS (Domain Name Server) query, connection establishment, waiting for the first byte, and downloading the page. The sources and potential candidates for bottlenecks are DNS query, server, links, and routers.

3.1 Non-Secured

This section examines the sources of latency in accessing non-secured web pages. We connect to various sites with image data in Europe and download files. We find that 80% time is needed to connect and only 20% time spent to download 1KB (normalized) file. The total time is broken down into four components: DNS query, connection setup time, time to get the first byte of a web page, and downloading time.

DNS Time: When a page is requested in the web, first the hostname is resolved to an IP address by DNS and then a connection is established from the client machine to the requested host. The DNS component measurement is the time spent resolving the DNS name to an IP address.

Connection Setup Time: The connection setup measurement is the time required setting up a connection from a client to a web server.

Time to Get the First Byte: The Time to Get First Byte measurement is the amount of time from when the client sends the request (GET command) until it sees the first byte back from the server.

Downloading Time: This measurement starts when the first byte arrives and ends when the last byte of the file arrives to the client.

From the moment the browser sends the acknowledgement completing the TCP connection establishment until the first packet-containing page content arrives contributes to roughly half of the time. The bulk of this time is the round trip delay, and only a small portion is the delay at the server. This implies that the bottleneck in accessing pages over the Internet is due to the communication delay and is not due to the server speed.

Bottleneck in DNS and Server: The timing in the four components were measured using Keynote's [16] tools. Keynote has seventy agents in the US and in selected foreign countries. The tools can measure the four components from the agents. one hundred sites suggested by PC magazine [23] were used to conduct this experiment. These experiments were conducted at Virginia Tech site because of availability of the permission to use the Keynote tools.

The average time for access from Paris to the hundred sites is 660 ms. The ranges of DNS time is 10%-25%, connection setup time is 20%-30%, time to get the first byte is 40%-60%, and of downloading time (1KB) is 10%-20%. The DNS time is significant in accessing international web sites. The variation in the range is due to acessing sites of different location of the world. Even though many sites are in the US, they are in sifferent coasts. Replicating the experiment several times saves DNS query time because data can be obtained from the cache. Statistics shows that cache hits save 90%-98% of DNS time. The Name Servers can be mirrored, especially for root and top-level domains to distribute them on the basis of geographic location to further reduce the DNS time. Around 40%-60% of total time is spent to get the first byte after the connection is established. A Bellcore web page [3,13] reported similar data and attributed it to the server delay. We conducted experiments to measure the server processing time. Different sites and file sizes are used for this experiment. The experiment was repeated several times and in most cases, the server delay was 1-3 ms. An artificial load on the server was imposed using Webjamma [25], an artificial HTTP traffic generator, to send a series of URLs to the server. The number of clients was increased such that the server can process 30 requests per sec. The CPU load was increased to as high as 100% by running a simple floating-point calculation program in the background. The CPU load was measured using *uptime* command of BSD for one min. The server response time did not go up beyond 3 ms. In one case, a 19 ms server response time was obtained when the server was 25% loaded. This is an exception.

The time to get the first byte from the client to the server needs some explanation. The client sends an ack of the server's SYN packet and waits for the first packet. The client needs to wait for a whole round trip time (RTT) and for the server's delay. Because the server delay is small (1-3 ms), the round trip time dominates the time to get the first byte. It is close to the connection setup time. Sometimes this time is high, which makes the overall time to get the first byte high. The time to get the first byte as a RTT by ping was validated. Ping was run while the server experiment was going on. In most of the cases, time to get the first byte is close to the average ping RTT.

Bottleneck in Link and Router The web page delay is dominated by RTT. Link and router characteristics have great impact on RTT. We setup experiments for links inside the US, from the US to foreign countries and vice versa to figure out what causes the RTT high. In each experiment, paths from a host connected to domain vt.edu (at Virginia Tech) by switched 10 Mbit/sec Ethernet to the remote servers are identified using pathchar [14]. End-to-end routing behavior in the Internet is discussed in [17].

To analyze RTT inside the US, the path for URLs of 80 universities and top 100 sites prepared by PC magazine were traced. The RTT ranges from 20-35 ms in case of East Coast sites, ranges from 40-55 ms in midwest, and of the ranges from 70-90ms in the case of West Coast from Virginia Tech university. In the east, the RTT is distributed all over the hops. None of the link/router dominates the overall RTT. But when the traffic goes to the West Coast, either the bottleneck is in Virginia Tech to vBNS (Very high speed Backbone Network Service) connection or in inside Sprint network. So, this link or the router on this route is a possible candidate for bottleneck when traffic goes from Virginia Tech to the West Coast .

When a packet goes to foreign country from the US, a high RTT is experienced inside Canada and the US, specifically just before leaving the country. This RTT is 40 times more than for the case of LAN. The standard deviation of RTT in this segment is very low, which means the segment often has this high RTT. For this purpose, the links that account for most of the high RTT (busy links) were identified.

There are some common characteristics of the busy links. For example, each link is inside Canada or the US and entirely in one network domain. The end routers that connect the busy link are considered in close physical proximity, and they are not connected by a satellite link. In most of the cases the first router of the busy link uses FDDI and the link is the final network link before traffic leaves the US or Canada to the overseas link. The causes of high RTT can be hypothesized as: A busy link router either has insufficient processing capacity or insufficient outgoing link capacity.

For incoming traffic to the US, remote country normally has lower bandwidth and introduces significant amount of delay. Traffic analysis from Bangladesh to the US shows that a very high latency is imposed by satellite connection (order of 750 ms) as well as the transoceanic link (order of 150 ms). So, the cause of high RTT for a incoming packet to the US can be summarized as follows:

1. Satellite connection anywhere in the path causes high delay
2. Link connecting two countries often experience high traffic and causes high RTT

3.2 Secured

E-commerce transactions require secured transaction using the Internet. It is important to know the overhead is added for the secure connection. One research question is how the security issue is being performed both on the client side and server side during a SSL (Secure Sockets Layer)

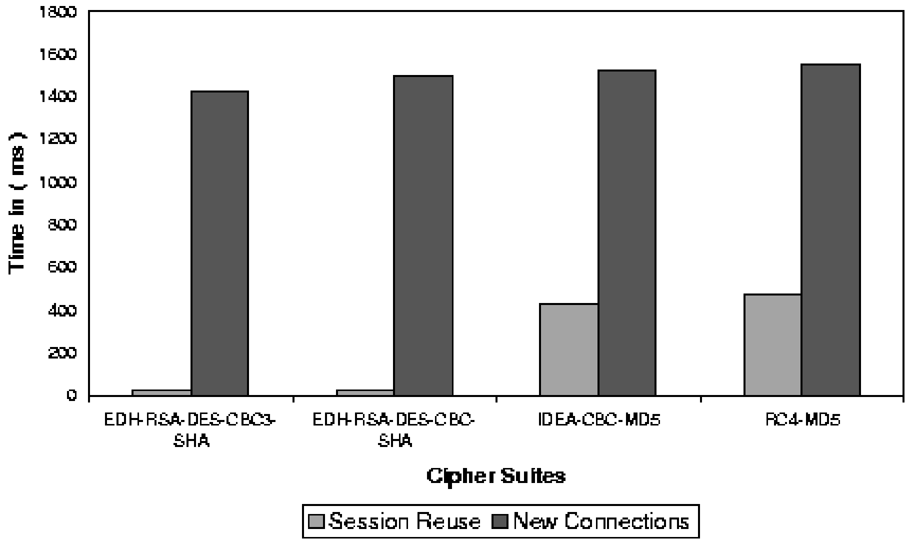

Figure 1: Performance measurements with new connections and reusing sessions

transaction. We consider the transaction initiated by a standard browser using non-proprietary technology. We used different variation of Secured Hash Algorithm (SHA) and Message Digest version 5 (MD5) as a cipher suites in our experiments. For details please see [18,21]

Details of a SSL Transaction: First the server and the client negotiates a master key, which both sides can use to encrypt/decrypt data transferred between the two during a session. The server proves its identity to the client by presenting a Digital Certificate signed by a Certificate Authority whom the browsers trust. This authority could be Verisign, AT&T, etc. Depending on the resource being requested by the client, the server may force the client to authenticate itself by producing a client side certificate. Most commercial sites augment their access control mechanisms by requiring an additional password from the client but that is done after the SSL transaction is already established. Once all conditions are satisfied and if the connection establishment phase passes successfully through the verification phases, both client and the server starts exchanging data. This data is encrypted with the key negotiated before until the connection is torn down.

Measurement Parameters: The objective is to measure the performance of the steps in an Internet transaction accross different parameters like:

- Selection of different Cryptographic algorithms suitable for local needs
- New sessions vs Reusing the session
- SSL version 2 Vs SSL version 3

Experimental Setup: In order to measure the security overhead of the transaction without accounting for the network delay or delay from other subsystems, measurements were done between SSL client and a server running on the same machine. The platform used for the test was SSLeay [22], which was installed on top of Solaris in a Pentium machine. Measurement scripts are written on top of it.

Discussion: The experiment shows that SSL version 2 is much faster than version 3 irrespective of the cipher suite being used. It shows that the strength of the crypto algorithms affects the performance. Figure 1 shows that reusing the same session to create a new session is much cheaper since the client/server does not have to go through the lengthy key exchange algorithm but can rekey their new session parameters from the context of the existing SSL transaction. The new session takes as much as 1.5 sec. This time is an extra overhead to setup connection.

4 Experiments on Digital Library Transactions

Web access of Digital library data involve transmitting text, image and various kinds of multimedia data over the Internet. A series of experiments on digital library transactions over web has been done using image data files [5].

The following subsections study the round trip time taken to transmit image files over the Internet. Similar studies in a LAN environment have been conducted for comparison. The time taken for lossy compression and decompression has also been studied, as well as the tradeoff possible between communication speed and quality.

4.1 Input Parameters

Nineteen NASA image data files were selected to measure communication times. The files range from 7K to 400K. The files were chosen so that they are representative of any image database in a digital library - they represent different image characteristics like color, shades, contours, texture etc. The size, resolution and other details of the images are given in Table 1.

When compressed files were required for the experiments, the JPEG compression method was used to get different levels of compression. The following notation was used: 10% quality level means that approximately 10% of the file has been retained.

4.2 Measurement of Communication Overheads with Image Data

Experiments were conducted to measure communication overheads using the connection-oriented transport service (TCP). The communication experiments have been divided into two parts: over the LAN and on the Internet. Measurements were made at different times of day to measure the communication delay due to network traffic.

Experimental Procedure: It is difficult to have computer accounts in more than one administrative domain [6]. Since we do not have an account on the remote host, a TCPecho program was designed which uses a TCP client to echo a file using port 7 on the specified remote machine. Timestamps are noted before sending the file and after it is echoed and received back. Each trial in the experiments consisted of 100 repetitions and each was repeated three times for a total of 300 measurements.

Experiment 1: Measurement of Communication Times in a LAN

Statement of the Problem: The purpose of this experiment is to measure the performance of communicating digital library data in a local area network (the remote site is only one hop away) and a metropolitan area network (the remote site is four hops away). This experiment was

SIZE	CONTENT
6988	**earth-round.gif:** Green on blue, sharp contours. Res (Resolution): 187x158
7708	**earth1.gif:** Green on blue, very sharp contours. Res: 160x160
17027	**gal_line.gif:** Red on black, a line of only dots. Res: 450x450
29668	**gal_green.gif:** Green on green, lots of dots, striations of colors . Res: 384x330
35543	**comet.gif:** White eye, blue tail, tail fades into background. Res: 512x480
60379	**mars.gif:** Huge circle of light brown shades. Res: 340x340
74058	**surface.gif:** White and blue shades, sharp contours. Res: 550x450
80385	**jupiter.gif:** Red and yellow shades, yellow text on black. Res: 710x765
97835	**gal_blue.gif:** Blue on blue, some dots. Res: 607x373
104365	**hubble.costar.gif:** Concentric red, orange, yellow shades, text. Res: 566x384
114323	**earth_detail.gif:** Pink on black, blurred contours. Res: 1152x864
135701	**eclipse2.gif:** A huge number of red shades. Res: 784x630
153634	**4gal_red.gif:** Bright red, orange; black and white dots; shading. Res: 441x400
175405	**sf.gif:** Sharp boundary contours, blue, white and red colors. Res: 500x500
205747	**ast_spray.gif:** Black background, lots of small particles. Res: 701x659
236199	**mitwave1.gif:** Delicate orange and white ridges. Res: 1024x1024
279786	**earth_highres.gif:** Blue on black, blurred contours, text. Res: 1152x864
406851	**text+image.gif:** Text, many dots, subtle shading. Res: 936x867
486430	**eclipse1.gif:** A huge number of orange and yellow shades. Res: 1280x1024

Table 1: Input files used in our experiments

performed using both uncompressed and lossy compressed files. This yielded a measure of the improvement of performance when a lower quality image was retrieved.

The experiments in the local area network were conducted between two Sun Sparc workstations **raid8** (Sparc 1) and **pirx** (Sparc 10) in the lab and **atom**, a machine in the engineering network at Purdue (ecn). Raid8 was the machine that was used to conduct the experiments and pirx and atom were used as receiver sites. The number of hops between raid8 and pirx is one and they are connected by a 10Mbps Ethernet. The number of hops between raid8 and atom is four.

Results: Figure 2 compares the round trip times of uncompressed and compressed files between two machines (raid8 and pirx) in a local area network. From a human perspective, the difference between compressed and uncompressed files, especially for small size files is not significant. Figure 3 compares the round trip times of uncompressed files in a LAN and a MAN.

Discussion: The files under observation range from 6 K to 248 K. In a LAN, the round trip times range from 722.84ms to 814.085ms. In a MAN, the round trip times range from 749.41ms to 2072.11ms. Two observations can be made here. The difference between a LAN and MAN for a file of size 6 K is only 26.57ms. Thus there is only few milliseconds increase in communication time when a file of that size is communicated through four hops instead of one hop. On the other hand, the difference in round trip times for file size 248 K is 1258.025ms. Thus in a digital library environment, small image files can be retrieved without lossy compression being performed on them to reduce quality of data. The second observation is that the difference in round trip times in a LAN environment between files of sizes 6 K and 248 K is only 91.245ms. The same difference in a

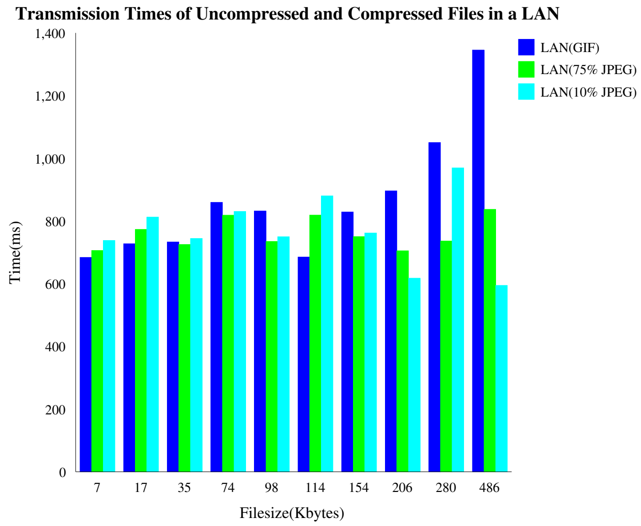

Figure 2: Variation of Transmission Time with File Quality in a Local Area Networ k

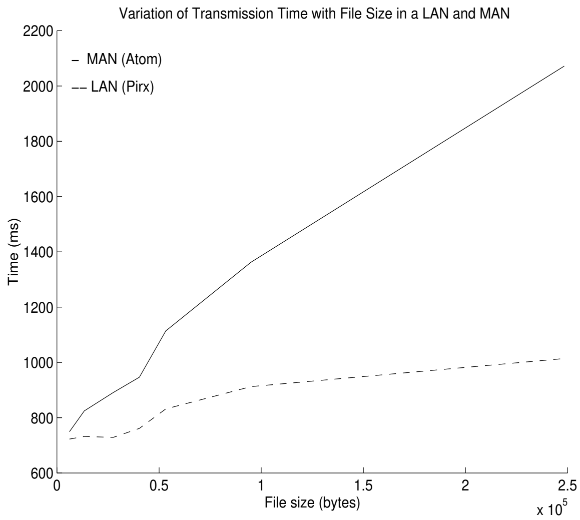

Figure 3: Variation of Transmission Time with File Size in a LAN and MAN

MAN environment is 1322.7ms. Thus in a MAN, the difference in communication times between the large file and small file in the sample is more significant than the corresponding difference in a LAN.

Figure 2 shows that there is no clear advantage of using a lower quality image file except for very large files. Using an uncompressed file is the same as using a compressed file. Thus in a local area network, large files can also be retrieved as they have been stored without any operations being performed on them.

Experiment 2: Measurement of Communication Times over the Internet

Statement of the Problem: The purpose of this experiment is to measure the performance of

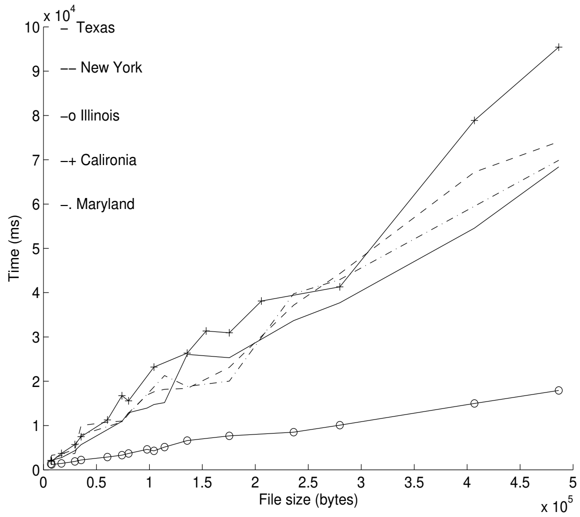

Figure 4: Variation of Transmission Time with File size in a WAN

communicating digital library data over the Internet. As in experiment 1, both uncompressed and lossy uncompressed files were used to observe the the improvement in performance with decrease in image file size by reducing image quality.

Procedure: The same TCPecho program was used as in experiment 1. This experiment used the following five remote sites. The approximate number of hops are also given:

- *Retriever.cs.umbc.edu:* (Maryland). Number of hops = 25
- *Bovina.cs.utexas.edu:* (Texas). Number of hops = 23
- *Lanai.cs.ucla.edu:* (California). Number of hops = 22
- *Ironweed.cs.uiuc.edu:* (Illinois). Number of hops = 19
- *Merope.cs.buffalo.edu:* (New York). Number of hops = 19

Results: Figure 4 shows the round trip times among the chosen five sites and raid8 at Purdue. The files used were GIF files. Figure 5 compares the the round trip time with the remote site in California when two JPEG quality levels are used - 10% and 75%. Figure 6 gives the comparison among a LAN site, a MAN site, and two WAN sites for files ranging from 6K to 250K. This figure shows that if the receiver sites were in the same local area network or within 3-4 hops (same MAN), there is only a one second difference in round trip times between the 10K and 400K file. On the other hand, the sites in Maryland and California result in more sharply increasing round trip times as the file size increases.

Figure 7 illustrates the difference in round trip times between night and day. The experiments were conducted at 2.00 am and 12.00 noon. It can be observed that the difference between the two curves widens as the file size increases.

Discussion: The round trip times rise sharply as file size increases in a wide area network. Figure 4 illustrates the difference between the largest file and smallest file (23811.691ms) for lanai, the site

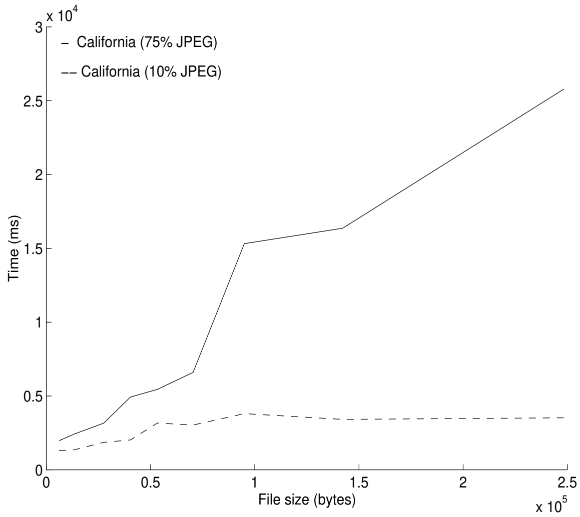

Figure 5: Variation of Transmission Time with File Quality in a WAN. Site: Lanai

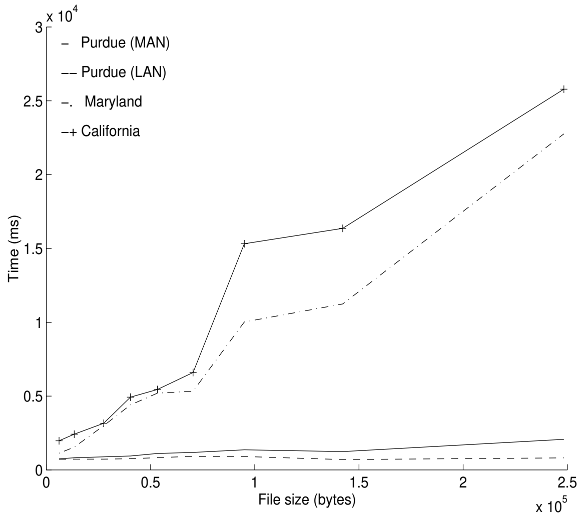

Figure 6: Variation of Transmission Time with File Quality in a LAN and WAN.

in California. This is much larger than the same difference in a LAN - 91.245ms. Figure 5 shows the round trip times for 75% JPEG and 10% JPEG files. For small files like 6 K, the difference in round trip times is 671.33ms and for a large file like 248 K it is 22264.111ms. Thus when a large size file is communicated by compressing it and losing some data, the improvement in performance is much more than when the size of the file is small.

As the number of hops in data transmission increases, it is more expensive to transmit data. For example, the round trip time to the site in California for a 250K file is 26 seconds. In comparison the couple of seconds spent for compressing and decompressing is a small percentage of the total response time. Thus Compressing and transmitting the file is worthwhile.

Experiments to Measure Compression and Decompression Time: The size of the

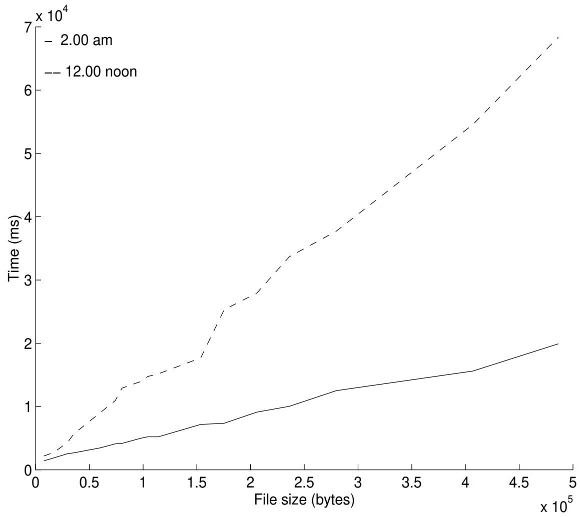

Figure 7: Variation of Transmission Time with File Quality with Time of Day

image can be reduced by lowering the quality of the image, as seen above. If a file is compressed using a lossy compression method and transmitted, then the response time consist of the time taken to compress the image at the sender's site, transmit the compressed file and decompress it at the receiver's site. To measure response time, experiments were conducted to measure compression and decompression time. The results are briefly printed here.

JPEG was used as the compression scheme and GIF files to represent uncompressed files. There is no clear correlation between the file size and the time taken to compress and decompress the file. The time taken for compression and decompression depends on the contents of the image like the number of colors, number of shades, amount of detail and so on. Using JPEG files can be compressed to different quality levels like 10%, 30%, 50% and so on. Different images have different quality levels at which they are visually indistinguishable from the original. The lower this level higher the reduction in response time. Thus the amount of data to be lost depends on the network distance, that is, how much the response time has to be reduced.

5 Experimental Setup for Web Transactions

In the past two years, a series of experiments has been conducted to study the performance of different kinds of web transactions. The ongoing experimental research in the Raid laboratory will pave the path for better understanding of the communication performance related issues on the web and suggest some concrete directions for tackling them. This section presents the apparatus for the experimentation. The next few sections give the results of the experimental study.

Experimental environment: This research is conducted in the Raid laboratory with access to Sun work stations The Raid Ethernet is a 10Mbps individual subnet of the Purdue CS departmental network. The connection from the Purdue–CS net to the Internet backbone is first through a NSC hyperchannel in Purdue campus, and then via a T1 line to the NSFNET T3 backbone. All Raid machines have local disks and are served by departmental file servers.

The Internet was used as the wide area network testbed. Internet connects over two million computers in over 50 countries around the world. The United States portion of the Internet is organized bottom up by linking many regional networks by high speed NSFNET backbones.

WANCE tool: A Wide Area Network Communication Emulator (WANCE) [26] tool was developed to *emulate* Internet communication in a LAN environment. It saves the researchers from the bureaucracy in different organizations/countries for conducting an experiment on actual geographically separated sites. The basic idea behind emulation is to divert communication between two local hosts to go through the real Internet. The emulation approach was chosen because it provides the real WAN communication in a LAN environment, capturing the dynamics of the Internet. As shown in figure 8, to emulate a three-site system linking A, X, Y on the Internet, The experimenter only needs to find two hosts B, C in the same LAN as A and *comparable* in performance to X, Y respectively. The transaction managers were run on the hosts A, B, C instead of hosts A, X, Y. The user of the WANCE tool can specify it to route all transaction packets from A to B through X, and A to C through Y. Thus, even though these experiments are running on the local hosts A, B, C (three work stations in the Raid laboratory), the same results as running them on the remote hosts X and Y are obtained. The justification for the emulation approach is based on the observation that the difference between the behavior of a distributed system running on a LAN and that on a WAN is primarily due to the communication performance. The validation of the WANCE tool has been studied and is found to report a deviation of $\pm 3\%$ compared to the *real* experiments on the Internet.

5.1 Experiments on Web Transaction Processing

The processing of web transactions involves a great deal of searching at the server sites. The performance of web transactions were studied, and the measurements obtained from these studies are presented here.

Statement of the Problem: Section 4, summarized that large variations in the communication performance metrics is a reality on the Internet. The time of the day and the location of the host are two of the main contributors to this variance. Since the transaction processing software components rely on the underlying communication software, it can be safely claimed that the performance of transaction processing will demonstrate a similar behavior. This experiment strengthens this assertion. It studies the effect of the time of day on the performance of web transactions

Procedure: The workload was a periodic batch of 20 web transactions every 10 minutes over a 24 hours time span. The 10 minutes interval between two batch submissions was small enough to capture the network dynamics. A smaller interval would unnecessarily clobber the Internet without contributing to the study. The batch of 20 transactions was large enough to generate meaningful average response time, throughput, and the abort pattern; but small enough for their execution (plus experimental setup and bookkeeping) to fit in the 10 minute interval.

The WANCE tool was used to conduct emulation experiments between Purdue and hosts in Germany, Finland, Norway, Israel, India, Japan, Hong Kong, Thailand, Australia, Brazil, Zambia, etc.

Data: The results of experiments between the hosts at Raid Laboratory and Helsinki (Finland); and between the hosts at Raid Laboratory and Hong Kong are presented. The choice of these

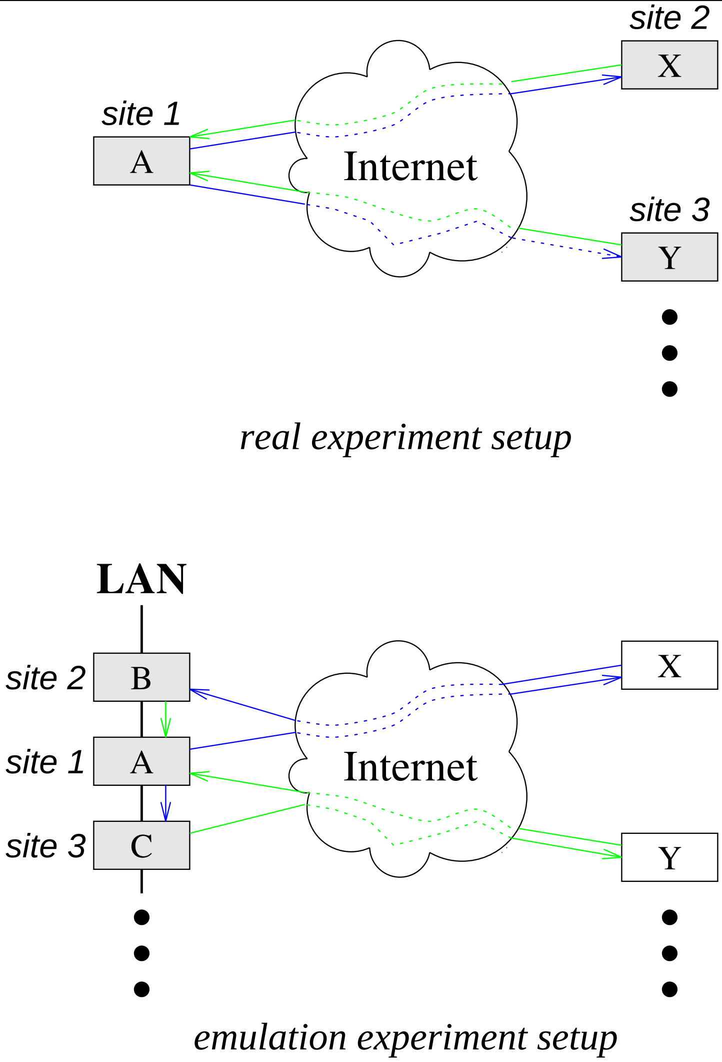

Figure 8: Configuration of emulation experiments

two remote sites can be explained as follows. First, the results from these two remote sites are representative of all the other test sites. And secondly, because they cover much of the global span and many different time zones, they provide a better understanding of how network metrics behave at different hours of the day. The response time, throughput, and transaction abort rate was measured for each transaction batch.

The results are graphically represented in Figure 9. $X-$axis represents the time of the day. The experiment was repeated over a period of 40 days. The data for each day was also averaged and the mean, variance, and the 95% confidence intervals for these averages were listed in this figure.

Discussion: The data shows that the performance of web transactions on the Internet can change drastically at different times of the day. For example, the transaction processing between Raid Laboratory and Helsinki (Finland) has much better performance at night than during the day. This can be attributed to the fact that the minimum time zone difference between North America

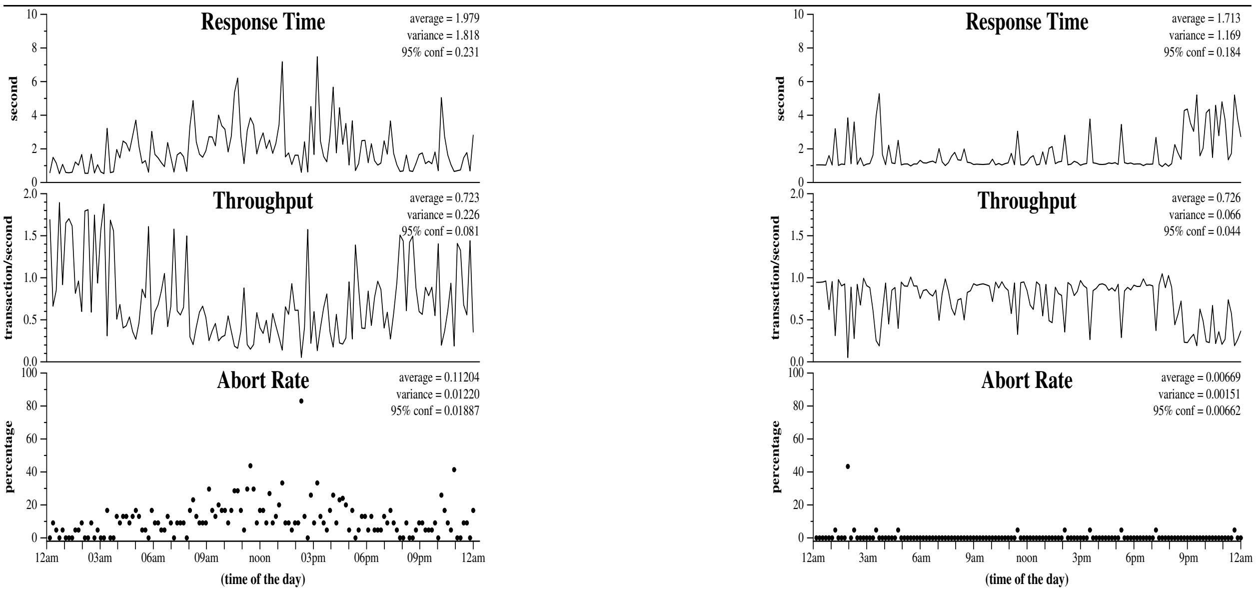

Figure 9: Raid Laboratory–Helsinki Raid Laboratory–Hong Kong

and Europe connected via a few trans–atlantic links is only about three hours. This suggests that the network traffic patterns at different times of the day are similar on at least these "shared" links. This is consistent with the previous studies that show the message delivery is slower and less reliable during the working hours in this region. The throughput was almost 2 transaction/second at night, but dropped to below 0.5 transaction/second during the day time.

The web transaction performance between Raid Laboratory and Hong Kong is better than the previous case. It is attributed to the following two reasons: First, the time difference between these sites is 13 hours. Thus, when it is day in USA, it is night in Hong Kong. Second, the US–Hong Kong Internet link is more reliable as has been evaluated in. Unlike the US–Europe Internet communication, where so many countries share a few cross-Atlantic links, the US–Hong Kong cross-Pacific link serves only Hong Kong, an area smaller than New York city.

6 Performance of Basic Database Operations in WAN

Statement of the Problem: A web transaction consists of basic operations like updating an existing disk block, inserting a new disk block, and selecting a disk block. Understanding the performance of basic operations helps to extrapolate the performance of a particular transaction given its access pattern. In this experiment, the elapsed time for each of these constituent operations in a transaction was measured.

Procedure: The cost of executing three basic queries was studied and analyzed: a query selecting one disk block, a query updating one, and a query inserting one disk block. Each disk block of the relation is 512 bytes long.

The system was configured to emulate a host in UTA (University of Texas in Arlington), a host in Hong Kong and a host in Finland. The same experiment was repeated in the LAN to get the data set for comparison. Measurements are repeated 100 times for each query (plus an extra first time to eliminate the cold-start effect). Query aborted due to the timeout because of long delay or message loss in a WAN, would be restarted later.

transaction	within LAN	Purdue–UTA	Purdue–Helsinki	Purdue–Hong Kong
retrieve one disk block	167	166	166	167
insert one disk block	286	352	754	1768
update one disk block	267	339	785	1786

Table 2: Response time (millisecond) for basic database queries

Data: Table 2 shows the time in millisecond taken by the processing of the several basic web queries. The time to retrieve one disk block is the same for different configurations. This could be attributed to the fact that all disk block reads can be completed using the local cache. Thus, the distance between two sites did not affect the performance of retrieval queries. For insert and update queries, the distance played a significant role in their performance. The time for inserting or updating one disk block in the Purdue–UTA case only increases slightly over the LAN, because the communication delay between Purdue and UTA is around 35 milliseconds (averaged), causing the processing time to dominate. However, in Purdue–Helsinki and Purdue–Hong Kong experiments, the response time is notably increased. Communication delay is around 300 millisecond (averaged), for the Purdue–Helsinki case, and around 1300 milliseconds (averaged) for the Purdue–Hong Kong case. Thus the message latencies dominate the processing times for these queries .

Next, the effect of varying the number of disk blocks in a query on its performance was studied. Thee response time of the queries for inserting or updating or selecting 10, 20, 30, or 40 disk blocks in a relation was measured. The results are plotted in Figure 10. (Only the data from the LAN is plotted for the retrieval queries because of their uniform performance in both LAN and WAN due to the replicated data.)

Discussion: The response time for each query is an approximate linear function of the number of disk blocks in it. Both update and retrieve queries are more sensitive to the number of disk blocks as they involve indexing, reading, and concurrency control overheads. Insert does not need such overhead. However, the gap between response time in a WAN and that in the LAN does not increase with the number of disk blocks. This implies that increasing the data size in a transaction will increase the response time in both LANs and WANs, but the decisive factor is the communication delay not the size of the message. This also suggests that it is possible to project the performance of transaction processing from LANs to WANs.

6.1 Multi-Programming Level and Concurrency Control

Statement of Problem: The number of concurrent transactions in a system at any time is called its multi-programming level (MPL). This experiment investigates the relationship between the MPL and the Web Transaction Processing (WTP) [8] performance on the Internet using different combination of concurrency and atomicity protocols. The two popular concurrency control and atomicity control protocols have been selected: two-phase locking (2PL) and timestamp ordering (TO); and two-phase commit (2PC) and three-phase commit (3PC) respectively.

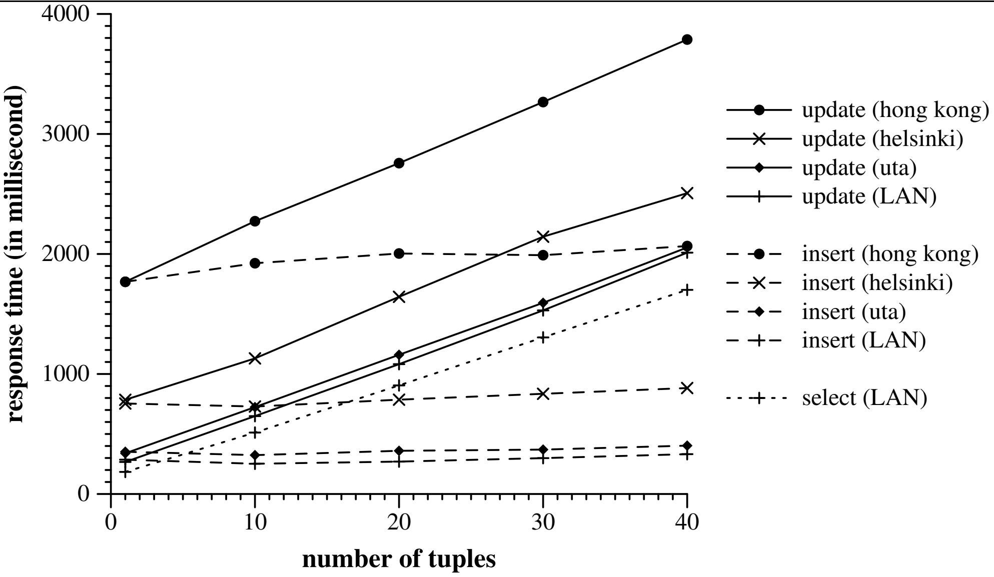

Figure 10: Response time for insert/update queries of several disk blocks

Procedure: The performance of WTP in three different site configurations was compared: within Raid Laboratory (LAN); Raid Laboratory and University of Texas, Arlington (USA); and Raid Laboratory and University of Melbourne (Australia). A workload of 250 transactions with 50% average updates is used. The MPL was varied as 1, 2, 3, 4, 5, 10, 15, and 20 (MPL=1 is the normal batch mode). The WANCE tool is used to emulate the Internet communication for distributed transactions. The performance of four different protocol configurations was measured: 2PL with 2PC, TO with 2PC, 2PL with 3PC, and TO with 3PC. For each value of MPL, the experiment was repeated 50 times and averaged the measured data.

Data: Figure 11 shows the response time, throughput, and abort rate of the benchmark transactions using different combinations of concurrency and atomicity control protocols, as the MPL is varied. The results for WTP in LAN (leftmost graph in fig 11) environment are included for the comparing the performance.

Discussion: As the MPL increases, the response time increases monotonically in all three configurations, which is as expected. The increase in MPL of the system implies more data item conflicts and hence the blocking/aborting of a transaction, delaying its completion.

The abort rate also rises with the increase in MPL in all the three configurations. In 2PL based combinations, the higher contention for data items increases the probability of deadlocks which is resolved by aborting the transactions. In TO based combinations, the probability of "out of order" arrival of read/write actions on a data item increases as the number of transactions in the system (MPL) increase. TO scheduler rejects each such action, aborting the transaction. The $2PL+3PC$ case shows the highest abort rate. This is probably due to the extra round of message required in the Internet for committing a transaction. This extra message round increases the lock holding time for a transaction which increases the probability of deadlocks. This might indeed be the case because a similar behavior is absent in the local LAN case.

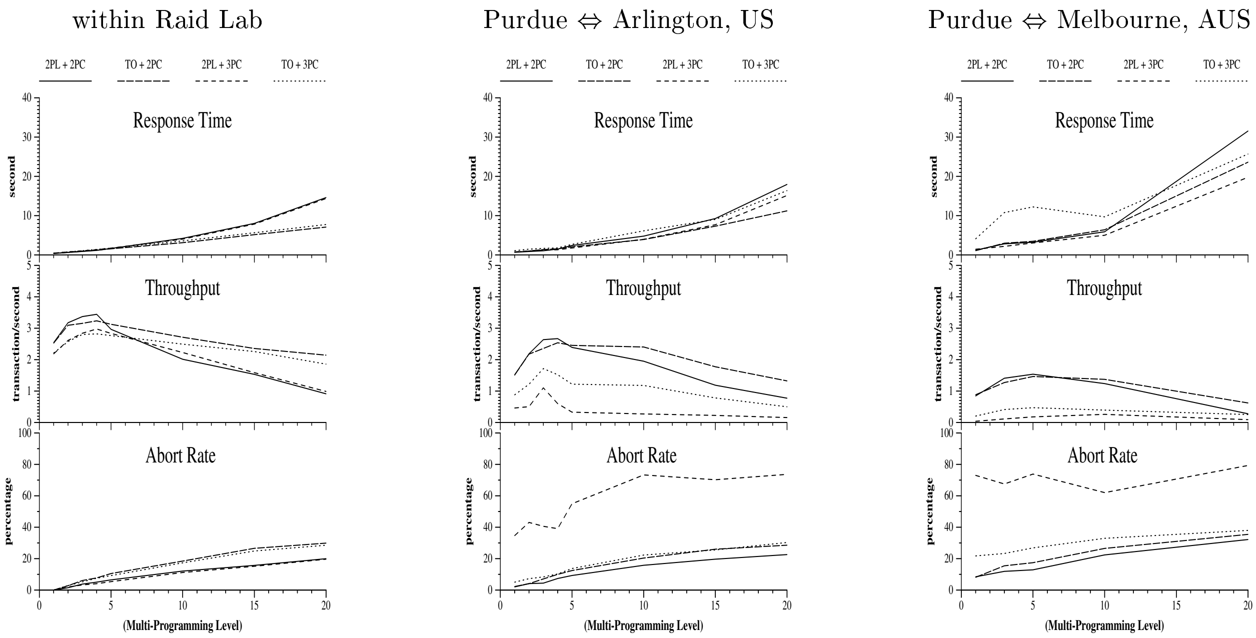

Figure 11: MPL experiments with LAN, UTA, and AUS site respectively

The throughput increases initially for small MPL, but starts to dip for larger values of MPL. The increased throughput is due to the "non blocking" concurrent execution of the transactions. As the MPL increases beyond a certain value, the transaction aborts increases due to increased number of conflicts and prevents any useful work to be done at the local site, thus reducing the system utilization.

The "best" MPL value is located in the range of 4 to 7 for each of the three configurations. This is the MPL at which the WTP system can provide acceptable response times without too much compromising on the concurrency, and hence the throughput. However, the throughput achieved at the "best" MPL value is lower and response times higher in comparison to LAN case. Furthermore, for MPL equal to one, the difference in the throughput between LAN and WAN cases can be factored out as entirely due to the message delays and losses.

The experiment suggests no significant impact of the MPL on the abort rate. This means that the unreliability of the Internet is the main contributor for the transaction aborts. In the LAN environment, the performance of TO scheme is inferior to the locking scheme. However, improved throughput with lower response times are observed in WAN environment with TO schemes. This preliminary result has been the motivation to look into the utility of locking schemes for the WAN environment more thoroughly.

6.2 Web Transaction Processing and Electronic Commerce Software

The architecture to set up the software for an electronic commerce systems is based on the client-server paradigm. A simple workstation or personal computer with internet connection serves as the client at the customer site. The window based interface with color can be useful. A large pool of webservers, a router with an incoming link to the outside world, a switch that routes the incoming package to a webserver and a large disk for the database are the main components at the vender site. A backend webserver software such as the "Apache" server or a dynamic content server can take care of most transactions and provides fairly enhanced capabilities. The database is usually a large depository of multi-media images including video, text, and files. The user interaction is through a window based system but eventually presented to the system in the form of database transactions. The book [15] discusses network infrastructure, architecture framework for electronic commerce along with issues of security and payments. The research work at Carnegie Mellon University [10] present the security issues, business models, set of protocols, and certified delivery mechanisms. A prototype implementation is also discussed. The research at MIT media laboratory [9] discuss the software agents that are intelligent versions of transactions and mobile processes. Software agents are employed to help with difficult and time consuming tasks in electronic commerce activity. The researchers have experimented with a prototype system called Kasbah. It was found that agents reduce transaction costs associated with end-consumer to end-consumer transactions where financial, timing, and trust issues can impede negotiations and commerce.

The core software in an e-commerce company is a distributed system in which the users transactions span over a wide area network. The system software/hardware of the vendor is usually in one location but proxy servers may exist at multiple locations to cache information needed by many users repeatedly. Even though some companies are providing dedicated communication lines (T1-56Kbps, T3-45Mbps lines) for large customers, much of communications takes place over the internet via Internet Service Providers (ISPs). ISPs offer high speed connections to Internet. Fiber optic-based integrated switching and transmission systems may allow for 155-1000 Mbps and may allow upto tens of millions of hosts in the future. Basically it is a local area network (LAN) environment in the customer and the vendor sites but they are conducting transactions via wide area network (WAN).

6.3 Electronic Trading Application on Web

In applications of electronic commerce there are many types of transactions. Examples of applications are financial institutions which can provide on-line access to status of accounts, orders, shipping date, pricing etc. Supporting payment by a client over the net brings the issues of security during the financial transaction.

Security can be enforced by authentication or encryption. Authentication has a communication overhead. It involves a lengthy exchange of information between the client and server such as keys before the secure channel is set up. Encryption has a computational overhead. If encryption is used only for small data messages used in a financial transaction, then the overhead is acceptable. But if huge multimedia data items are encrypted, along with the compression and decompression routines the encryption and decryption routines add a huge overhead to the data retrieval process [8].

Electronic trading is the most exciting but financially appealing applications. In electronic trading, there are several overheads. They are computation of algorithms, particularly encryption; I/O time for database access from various files; and the communication time for servers are involved in executing an order. For example, if the user wants to buy a particular stock, the real time quote has to be provided by the quote.com service. The user has to go the server at her machine to a server at the broker's site and finally to a server that has the actual information. In some cases, there could be as much as twenty message exchanges involved due to the additional need for authentication, reconfirmation, and seeking input from the user.

The process of executing a trade electronically is very similar to the process of trading in person or via phone. In our experience to trade on phone, we found the steps as follows. The person dials the phone number of the broker (phone may be busy), the person picking up the phone has to page the broker assigned to the account, the caller is identified (not much problem of security on phone). The customer specifies the stock of interest, asks questions like bid and ask price (detailed questions as volume, high/low of the day may not be always possible without being put on hold again). The customer places the order for the trade and the broker calls back with confirmation of execution. The steps take about 3 to 4 minutes and the return call from broker may take up to 15 minutes or more.

The electronic trading over the Internet is the emerging technology and over 20 percent of the trades are done via the Internet. Ameritrade claims to have three million accounts for electronic trading. It costs about $250 to set up an account. This is different than computer trading by institutional investors where computer sell/buy programs are triggered based on some criterion. The electronic trading involves distributed processing and communication among several servers. The network latency plays a major role in the response times. First one must open the browser such as Netscape or the Internet explorer. This takes about 15 seconds on a PC. Next the user accesses the broker's home page that takes another 2 seconds. It takes another 30 seconds to go the trading page where one can be secure and login. After logging, the customer may want to get real time quote from a New York Stock Exchange server (such as quote.com service). It has taken 5 to 10 seconds depending on the time of day. After the transaction is entered, the system presents the order back again to the user for confirmation and that takes about 10 seconds. The user finishes the transaction with a confirmation entry. The user can also check the status of the order in 5 seconds. Other requests for holdings in account, price charts, research reports are simple database queries and take 5-10 seconds.

The whole process of ordering a transaction for stock or option trading takes several round trips among the servers on the person computer, the broker's computer and the NYSE/NASDAQ computers. The communication time over the WAN, LAN, and the security mechanisms affect the response time for the customer. Some time is for the display of pages on the screen. If the communication time can be reduced, the transaction can take place in about two minutes. The stock price can fluctuate in this time so for a day trader, or institutional investor, this is too much time. The electronic broker can not succeed unless the communication behavior can be improved via higher bandwidth. The multi-media presentation requires better connections and modems. The communications will make or break the success of not only trading but other electronic commerce applications.

7 Conclusion and Future Work

It has been observed that communication cost is a dominant factor and a major performance bottleneck for the web transactions. There are a number of ways which can be employed to improve web performance. One of the most effective ways of reducing the communication delay is trough web caching. The documnets can be cached at client sites as is done currently by most web browsers or by web servers themselves, which is most useful when a server contains many pointers to other servers.

To reduce the communication delays that hamper tight interaction between software systems, it is appropriate to investigate the use of mobile software agent technology. This technology allows an agent in the form of program code, data and execution state to be packaged into a message and sent across the network to a remote computer. The agent can control multiple interactions with software resident on the remote computer to achieve the intricate negotiation or planning that is requied.The initial delay of sending the agent across a wide-area network will be longer than for a typical interaction message, because the software agent is larger than a typical message, but that delay is incurred only once. For activities that require hundreds or thousands of interactions, the time savings can be substantial.

For images, the option of reducing the size of the data without losing the semantic content has been explored. Beyond a certain threshold of network distance and file size response time is reduced if the image is lossy compressed and transmitted.

For the web page downloading, reducing DNS time will save overall downloading time. To reduce DNS time, the number of cached entries can be increased in local DNS server. Cache entries for popular servers can also be forced. The next source is connection establishment time. If persistent-HTTP is used, then connection time for more than one file can be saved. High RTT causes connection time, time to get the first byte, and downloading time high. The RTT depends on Link and Router. Link bandwidth is increasing day by day. This study shows that router is a great source of bottleneck both for inside the US and international countries. To increase the performance of router, active networking is being studied.

This study should give a deeper insight into the issues related with the performance of web transactions and will help to build more efficient, reliable and scalable systems.

8 Acknowledgements

Melli Annamalai contributed to Digital Library, Ahsan Habib contributed to the Web page downloading, Yong Zhang contributed to WANCE tool measurements, Pinkesh Shah contributed to security overhead, Angali Bhargava contributed to Qualiy of Service & electronic commerce trading applications.

9 References

1. M. Annamalai. Efficient Retrieval of Images in Distributed Digital Libraries, Ph.D. thesis, Department of Computer Science, Purdue University 1997.

2. B. W. Abeysundara and E. Kamal. High-speed local area networks and their performance: A survey. *ACM Computing Surveys*, 23(2):221–264, June 1991.

3. P. Barford and M. Crovella. Measuring web performance in the wide area. Computer Science Department, Boston University. Submitted for performance review. March 1999

4. A. Bhargava and B. Bhargava. Measurements and Quality of Service Issues in Electronic Commerce Software. *In Proceedings of IEEE Conference on Application-Specific Software Engineering and Technology (ASSET-99)*, Dallas, Texas, 1999, pp. 28-37

5. B. Bhargava and M. Annamalai. A Framework for Communication Software and Measurements for Digital Libraries. *International Journal of Multimedia Systems*, Vol 10, Jan 2000, pp. 205-235.

6. B. Bhargava, S. Goel and Y. Zhang. A Study of Distributed Transaction Processing in an Internetwork. *In Proceedings of the International Conference on Computer Information Systems and Managements of Databases (CISMOD)*, Bombay, India, Nov '95, pp. 135-152.

7. B. Bhargava, E. Mafla, and Y. Zhang. Evolution of communication systems for distributed transaction processing in Raid. *Computing Systems*, 4(3):277–313, Summer 1991.

8. B. Bhargava and C. Shi. A light-weight MPEG video encryption algorithm. *In Proceedings of the International Conference on Multimedia Information Systems (MULTIMEDIA 97)*, New Delhi, India IETE Jan 1998.

9. A. Chavez, D. Dreilinger, R. Guttman and P. Maes. A real-life experiment in creating an agent marketpalce. *In Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAMM-97).* April 1997.

10. B. Cox, J. Tyger and M. Sirbu. Netbill security and transaction protocol. *In Proceedings of the First USENIX Workshop on Electronic Commerce*, pages 77-78, July 1995.

11. R. Goldberg and D. D. E. Long. Accessing replicated data in an internetwork. *International Journal of Computer Simulation*, 1(4):347–372, December 1991.

12. A. Habib and M. Abrams. Analysis of Sources of Latency in Downloading Web Pages *In Proceedings of WebNet 2000*, San Antonio, Texas, Nov 2000.

13. C. Huitema. *Internet Quality of Service Assessment.* ftp://ftp.bellcore.com/pub/huitema/stats/quality_today.html, February 8 1999.

14. V. Jacobson and C. Pathchar. ftp://ftp.ee.lbl.gov/pathchar.tar.Z.

15. R. Kalakota and A. Whinston. *Frontiers of Electronic Commence.* Addisson Wesley, 1996.

16. Keynote Systems Inc. http://www.keynote.com, 1998.

17. V. Paxson. End-to-end routing behavior in the Internet. *In Proceedings of ACM SIGCOMM '96*, Palo Alto, CA, August 1996.

18. B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc. 1996

19. A. Z. Spector. Communication support in operating systems for distributed transactions. In *Networking in Open Systems*, pp. 313–324. Springer Verlag, August 1986.

20. S. Spiro. Analysis of HTTP performance problems. http://www.w3.org/Protocols/HTTP/1.0 /HTTPPerformance.html, July 1994

21. SSL Cipher Suites http://developer.netscape.com/docs/manuals/security/sslin/contents.htm

22. SSLeay source and documentation http://www.openssl.org

23. Top 100 sites from PC Magazine. http://www.zdnet.com/pcmag/ special/web100/index.html, 1998.

24. J. Touch, J. Heidemann, and K. Obraczka. Analysis of HTTP performance. USC/ISI, June 1996

25. *Webjamma.* WWW Traffic Analysis Tool, http://www.cs.vt.edu/ chitra/webjamma.html.

26. Y. Zhang and B. Bhargava. WANCE: A wide area network communication emulation system. In *Proceedings of IEEE Workshop on Advances in Parallel and Distributed Systems*, pp. 40–45, Princeton, NJ, October 1993.