Multimedia Tools and Applications, 10, 205–236 (2000)
© 2000 Kluwer Academic Publishers. Printed in The Netherlands.

A Communication Framework for Digital Libraries

BHARAT BHARGAVA bb@cs.purdue.edu
Department of Computer Sciences, Purdue University, W. Lafayette, IN 47907, USA

MELLIYAL ANNAMALAI mannamal@us.oracle.com
Oracle Corporation, One Oracle Drive, Nashua, NH 03062, USA

Abstract. Digital libraries involve various types of data like text, audio, images and video. The data objects are typically very large and of the order of hundreds and thousands of kilobytes. In a digital library, these data objects are distributed in a wide area network. Retrieving large data objects in a wide area network has a high response time. We have conducted experiments to measure the communication overhead in the response time. We have studied the correlation between communication and size of data, between communication and type of data and the communication delay to various sites in a local and wide area network. We present different strategies for reducing delay while communicating multimedia data. Images are amenable to losing data without losing semantics of the image. Lossy compression techniques reduce the quality of the image and reduce the size leading to a lower communication delay. We compare the communication delay between compressed and uncompressed images and study the overhead due to compression and decompression. We present issues in providing digital library service to mobile users and discuss a question: What if communication were free? Finally, we present a framework for efficient communication of digital library data.

1. Introduction

Digital libraries provide online access to a vast number of distributed text and multimedia information sources in an integrated manner. Providing global access to digitized information which is flexible, comprehensive, and has easy-to-use functionalities at a reasonable cost has become possible with the technical developments in several areas of Computer Science such as databases, communications, multimedia and distributed information systems. 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. Digital library applications basically store information in electronic format and manipulate large collections of these materials effectively. Their services provide capabilities for storing, searching, transmitting, viewing, and manipulating complex information. They provide a consistent and transparent view of underlying heterogeneous repositories.

Digital libraries typically deal with enormous quantities of data. The National Aeronautic Space Agency (NASA) has multiple terabytes of earth and space science in its archives. NASA is going to launch the Earth Observing System (EOS), a collection of satellites to be launched in the near future which will collect a terabyte a day. After 15 years, such an EOS database will contain 10^{16} bytes (10 petabytes) [17]. The Department of Defense has decades of data archives. Video-on-Demand systems have thousands of video clippings.

Almost every organization has repositories of old versions of software and business related data. The CORE project, an electronic library of Chemistry journal articles deals with 80 Gbytes of page images [7]. The University of California CD-ROM information system in 1995 consisted of 135 Gbytes of data [14]. It contains federal statistics such as U.S. census data, demographic information, maps etc. The ACM digital library, functional since July 1997, provides access to about 9,000 full text articles and several tables of content pages and bibliographic references.

Recent interest in digital libraries fuelled a stream of research, the most notable being the NSF-ARPA-NASA initiative sponsored projects in six universities [9, 16]. The University of Illinois project has focused on providing integrated access to diverse and distributed collections of scientific literature. They deal with heterogeneity, interfaces to multiple indices, semantic federation across repositories and other related issues. The group at University of California at Berkeley is working on providing work-centered digital information services. The issues they are involved with are document image analysis, natural language analysis, and computer vision analysis for effective information extraction, user interface paradigms, and client interaction with repositories. The initiative at Carnegie Mellon University is to build a large on-line digital video library featuring full-content and knowledge-based search and retrieval. The issues in this area are speech and image understanding, natural language processing, indexing based on content. The University of California at Santa Barbara has concentrated on geographical information systems. Their digital library project comprises of a set of Internet nodes implementing combinations of collections, catalogs, interfaces and ingest facilities. Finally, the Stanford University project addresses the problem of interoperability using CORBA to implement information-access and payment protocols.

1.1. Digital libraries in a distributed environment

Digital libraries are distributed over national and international networks and their infrastructure is inherently distributed [5]. Existing repositories are distributed, and the data needs to be shared by many users. Information processing is distributed, and particularly, the user's queries can be so complicated that the process of information retrieval requires multiple rounds of interactions between users and the various servers of the information system. These factors result in communication support being one of the most important components of any digital library architecture. Along with other components it contributes to the cost of providing digital library services. To keep the cost reasonable a digital library designer has to be aware of the communication overheads and the possible solutions to reduce these overheads.

In a wide area environment, the anamolies (failures, load on the network, message traffic) affect the communication of data. The multiple media of digital library data introduce further complexity since each media has its own communication requirements. The current network technology does not provide the bandwidth required to transmit gigabytes of digital library objects. The cost of access in the context of communication and networking is the response time required to access digital library data. A digital library user might have to wait for several minutes to receive the data due to bandwidth limitations.

We study communication in a distributed digital library at the information systems layer. The underlying information transfer mechanisms can be information protocols such as Z39.50 or HTTP. Our conclusions are general and apply to any underlying information retrieval protocol.

1.2. Magnitude in a digital library

The magnitude of the different digital library components contributes to the complexity of the digital library operations. Existing solutions have to scale up several orders of magnitude. We describe some of the problems caused by the need to scale up with emphasis on the communication requirements:

- Size of data: Digital library data objects can be very large—a compressed video file can be easily 500Mb. 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: Not only is each data object large, there are billions of such objects. A NASA image database would contain millions of images. A video database associated with a video conference would contain hundreds of thousands of video clippings. The library of Congress has thousands and thousands of books and journals. Query processing in a distributed environment containing these data objects will have expensive communication requirements.
- Number of sites: The number of locations of information repositories available is increasing everyday. This can be observed by the enormous increase in World Wide Web servers in the past year. A digital library database requires access to many more 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. In such an environment resource location and consistency management are communication intensive.
- Number of users: The Global Information Infrastructure visualizes every home in the world with access to a computer with easy access to the information highway.

1.3. Motivation of the paper

The motivation behind this paper is to investigate a wholistic approach to communication issues in digital libraries. Researchers have studied individual problems in the area of communications, but a more comprehensive approach is required because:

- Solutions to communication issues in different modules of a digital library affect each other and have to be studied as a whole.
- Digital libraries serve a diverse group of users and applications and a particular solution might not be relevant to all users and applications. The possibility of incorporating several different solutions has to be studied.

1.4. Organization of the paper

The goal of this paper is to identify the communication overheads in digital library transactions, investigate possible solutions and present a framework for efficient communication support in a digital library application. In Section 2 we describe the characteristics of digital library data media. In Section 3 we study in detail the various communication requirements of a digital library and solutions specific to the requirement and the media involved. In Section 4 we briefly study the communication issues in transmitting continuous data. Section 7 defines quality of service parameters for digital libraries and Section 8 presents our communication framework. Section 10 discusses the issues in providing digital library services to mobile users followed by the section on the futuristic question: what if communication were not an issue at all?

2. Media of digital library data

One of the factors that communication cost depends on, is the media of the data item transmitted. Digital library data item can be photographs, video clippings, dictionaries (which include both text and pictures), slides, speeches, music, medical scans, X-rays etc. We categorize these data items as belonging to four different kinds of media: text, images, video, and audio. The communication requirements of each digital library media is highly dependent on the structural nature of the data. The issues involved in the communication of data cannot be studied independently of the nature of the data and requirements based on the structure of the data. Hence, in this section we will briefly describe the characteristics of the different media. This helps us understand the specific relationships each media has with the communication infrastructure.

Text and image media are discrete media while video and audio are continuous media. The communication cost varies for data items of discrete and continuous media. Continuous media data items can be treated as a series of discrete data items with respect to some issues. We first study discrete media data items and in a later section identify the differences and ensure that the solutions we suggest take advantage of the knowledge of the nature of the media.

Digital library data items can belong to either one media only or be a combination of different media. For example, a dictionary will contain both text and images. A video clipping will contain video, audio and text-based annotations. We briefly study the effect combining two or more media has on communication in a later section.

2.1. Size of digital library data

The large size of the digital library data objects is one of the causes for the communication costs. In Table 1, we give a few estimates of the size of digital library data objects to give an idea of the order of size a digital library application deals with. The figures do not represent an average or generalized size of data items of a particular media, but a sample of possible data item sizes.

Table 1. Examples of digital library data item sizes.

Media	Size (Mbytes)
Text (An encyclopedia section)	0.1
Image (A NASA image)	0.8
Video (Uncompressed)	48000
Audio (2 minute speech)	1

Compression has been used to reduce the size of data items and storage requirements. It can also be used to decrease communication cost. Compression cost and applicability to realtime applications are factors which impact using compression to reduce communication cost.

2.2. Text data

Text data can be NBA scores, weather reports, software, dictionaries etc. Text is typically represented using ASCII, which results in a series of characters each represented by a byte. Higher level text data structures such as documents are highly structured.

2.2.1. Text data compression. Text data can be compressed using simple techniques and the compression ratio ranges from $2:1$ to $4:1$ [10]. Text data cannot afford to have loss, hence lossy compression techniques cannot be applied. Unlike image or video data, loss of even a small percentage of the data could lead to incomprehensible data. For instance, loss of a single line in a program will render it unusable. If a few letters in a sentence are lost, the dictionary entry will be useless.

2.3. Image data

An image refers to a 2-dimensional set of data points conveying visual information. The data points are *pixels* and the values represent the gray scale or color intensities of the corresponding pixel. In color images each pixel has individual components depending on the color model. 24-bit color images using the RGB color model have 8 bits each for the red, green, and blue components. Image data is used in several applications such as remote sensing, medical imaging, fingerprint databases, news databases, geographical databases, and art databases. Digital library applications involve image repositories such as encyclopedias, news photographs, NASA data (such as the Mars Pathfinder data), classroom slides, etc.

Image files sizes range from less than 100 Kbytes to over a 1000 Kbytes. A thumbnail sized image (80×80) will be around 100 Kbytes, while a large NASA image will be around 1000 Kbytes, satellite images can be in the order of megabytes.

2.3.1. Representing images. There are a wide variety of image file formats in use now to represent images [15]. The formats vary based on what they can provide in terms of quality, flexibility, computation, storage, and transmission efficiency. Some popular image formats suitable for UNIX platforms are BMP, Sun rasterfile, XBM, XWD, TIFF, DXF, GIF, JPEG, MPEG and UNIX Plot. Formats commonly found on PCs include PPCX, TIFF, IFF, IMG, TGA, DXF, HPGL, PCL, PIC, and CGM and on the Macintosh one can find MacPaint, PICT, TIFF, CGM and QuickTime RLE [15].

2.3.2. Image data compression. Images can be compressed. We study compression schemes which can be broadly classified into two categories: dictionary based schemes and transform based schemes. The two categories have different communication requirements and we will describe them briefly here.

Dictionary Based Schemes: Dictionary based schemes are based on the idea of having a lookup table containing data blocks followed by a data stream with pointers to the entries in the data block. Every data block in the data stream is replaced by a pointer to the location of that data block in the lookup table. The size of the pointer is assumed to be less than the data block, and also several data blocks might be identical and so require only one entry in the lookup table. Various adaptations of the scheme differ in the way the lookup table is created. They can be broadly classified into static and dynamic schemes.

GIF is an example of a file format which uses a dictionary based scheme for compression. This results in a sequential organization of a GIF file consisting of the header block, logical screen descriptor block, global color table block, image data blocks, and a trailer block. The sequential organization makes it a convenient format for use in communication.

Transform Based Schemes: Transform based schemes convert the data into another domain. For instance, the Distance Cosine Transform (DCT) converts the spatial image representation into the frequency domain. In this domain, the data can be “smoothed” (high and low points are smoothed off) resulting in loss of some data. This leads to lossy compression schemes where, with little or no visible degradation, and a high compression ratio can be achieved.

JPEG is a compression scheme that uses DCT. JFIF is a format which uses JPEG compression. The JPEG scheme consists of the following steps: downsampling to exploit the human eye’s lesser sensitivity to chrominance channels, DCT transform application (high frequency data are lost after this), quantization, and final encoding of the resulting coefficients. The encoding and decoding are more complex than in dictionary based schemes.

2.4. Video data

Video data can be considered to a be a series of images taken at subsequent intervals in time. Digital video data is obtained from analog signals by sampling the signal at a chosen number of sampling points. The minimum rate at which the analog signal must be sampled is twice the highest frequence in the signal. For NTSC systems this rate is $2 \times 4.2 = 8.4$ MHz and for PAL this rate is $2 \times 5 = 10$ MHz. Typically higher sampling rates are used for ease of signal recovery.

2.4.1. Representing video data. Analog video is represented using NTSC, PAL or SECAM formats. Video is digitized using the operations of prefiltering, sampling, quantization, and encoding [11] and represented using CCIR-601 (the digital television standard), Source Input Format (SIF) or Half Horizontal Resolution (HHR). Video for entertainment television in North America has about 30 frames per second, and about 480 scan lines per frame.

2.4.2. Video data compression. MPEG and MJPEG are the two most popular compression schemes for video data. MPEG uses an asymmetric compression method: compression is more computationally complex than decompression making it suitable for archiving systems and real-time systems. MPEG uses two types of compression methods to encode video data: interframe and intraframe coding [15]. Interframe coding is by encoding only differences between a frame and the previous frame and not the entire frame. This idea of predictive coding can be extended to bi-directional coding where the differences between the current, previous and next frame are encoded. MPEG supports interframe and intraframe coding by using three different types of encoded frames: I-frames (intraframe encoded), P-frames (predictive encoded), and B-frames (bi-directional encoded). The added features include interlaced video formats, multiple picture aspect ratios, conservation of memory usage, and increased video quality.

MJPEG (Motion JPEG) is a simpler encoding scheme which compresses each frame using JPEG compression method. The advantages are that it is fast with a real-time compression rate. The disadvantages are that there is no interframe compression leading to low compression ratio (MPEG can be upto three times better than MJPEG). Their large size also leads to a slower playback time.

2.5. Audio data

Digital audio data is obtained from the analog representation by sampling the analog data for voltage changes. The total number of bits required to encode one second of sound equals the number of samples taken per second (sampling rate) multiplied by the number of bits used to record each sample (sample size). The sampling rate and the sample size determine the quality of the digitized audio. Typically the frequency of audio supported is half of the sampling rate. Thus, voice data is sampled at 8,000 Hz to encode frequencies upto 3,400 Hz. Higher quality audio data such as music on CDs is sampled at 44,100 Hz to encode all frequencies upto 20,000 Hz, the highest frequency humans can hear.

If 16 bit samples of CD quality data were recorded using the heuristic of having a sampling rate twice the actual audio bandwidth (that is 44,100 Hz), one second of music would be around 1,400 Kbytes. If voice were recorded using 8 bit samples, one second of voice would be around 8 Kbytes.

2.5.1. Representing audio data. There are several audio formats in use. Some of them are: U-law, WAV, AIFF, and MPEG. MPEG is a standard and achieves better compression ratio than the other formats.

2.5.2. Compression of audio data. As with other media, there are both lossless and lossy compression schemes for audio data. GSM (Groupe Speciale Mobile) is a lossy compression scheme focused on speech. GSM uses a variation of Linear Predictive Coding (RPE-LPC: Regular Pulse Excited—Linear Predictive Coder with a Long Term Predictor Loop). GSM compresses 160 13-bit samples (2080 bits) to 260 bits which is an $8:1$ compression ratio. For 8 KHz sampling, this means that GSM encoded speech requires a bandwidth of 13Kbps. GSM is computationally expensive; if it has to be used for real time communication, a medium scale workstation is required.

Pulse code modulation. This family of encoding methods are applicable to any kind of sound data, that is, they are not restricted to either speech or music. Each sample is represented by a code word. The quantizer step spacing is transformed to logarithmic allowing a larger range of values to be represented by the same number of bits. 8-bit samples can be used to represent the range of values that can be achieved by using 14-bit samples resulting in $1.75:1$ compression ratio. The transformation to logarithmic values enables low amplitude samples to be encoded with higher accuracy than high amplitude samples. u-Law and A-Law are some examples of transformations. In uniform PCM methods audio samples are encoded independently from adjacent samples. Adaptive Differential Pulse Code Modulation (ADPCM) methods utilize the fact that adjacent samples are similar to each other and that the value of a sample can be predicted using the value of adjacent samples. Instead of quantizing the original samples, like the uniform PCM methods, ADPCM quantizes the difference between the original and the prediction. This results in accurate representation with lesser bits if the prediction is accurate.

We believe that audio data, unlike video or image data, can tolerate less loss. One might watch a movie with a blurry screen but will not listen to music on a radio with a lot of disturbance. The loss tolerated can also vary with the content of audio data. More loss can be tolerated in a speech than in music. Disturbance in a politician's speech might be acceptable when compared to a sports commentary.

3. Communication in digital libraries

Digital library application have the following modes of operation: querying, browsing, data retrieval, and interactive sessions. The communication requirements of each of these sessions is different. In this section we describe each mode, evaluate the communication overheads, and investigate solutions for each mode of operation individually.

3.1. The query phase

When a user wishes to use a library, digital or otherwise, the purpose might be to merely browse through the collections available or if she has a specific data item in mind, execute a query for the location of the data item by presenting a query. The situation might be somewhere in between the two, where the user has some idea of what data item she wants but is unable to completely specify the data item. In such cases she might want to process a query to filter out some data items which are obviously not what she is looking for and then browse through the rest. We will first discuss the query phase and in the next subsection the browsing phase.

The structure and size of a query depends on the media, and the precision of the user's request. A query can be keyword based for all media, and visual based for images and video data. An imprecise query specified might be specified in a natural language and has to be parsed before it is presented to the system.

3.1.1. Queries for text data. If the text data is represented using the relational model the queries are SQL-like declarative queries. A simple query based on the select statement could have a size of 100 bytes. If the text data is represented using an information retrieval document model, the queries will be in the form of boolean statements and will be 100 bytes or less in size. If the query is interactive, there are several queries with a similar size.

3.1.2. Queries for image and video data. Queries on image and video data are based on the features extracted from the data. Some examples are "retrieve all images with 60% blue color in the upper half" (perhaps to retrieve all images with a blue sky), "retrieve all images with 10% black and texture similar to the human skin" (perhaps to retrieve all images with a human wearing a black suit). Queries for visual data can be keyword based or visual based:

- Keyword based: These queries are similar to the text data queries described above. They can be specified in a text file or using a menu.
- Visual based: Visual queries include an actual visual data item. Examples of the form of such queries are "retrieve all images similar to this image" or "retrieve all video clippings which contain the person in this image". If the user uses an abstract, high-level version of the data, such as a thumbnail version of the image, then the query size will be of the order of a few tens of kilobytes. An 80×80 image will be about 30 Kbytes if each pixel is represented by 24 bits. If a full size image is used in the query, then the size of the query increases anywhere from 100 Kbytes to 1000 Kbytes or even more depending on the size, resolution, and compression scheme used for the image.

3.1.3. Queries for audio data. We believe that queries for audio data will be keyword based. It is possible that future applications will take as input a voice or music sample and retrieve all audio data containing that voice sample. In this paper we assume that all queries for audio data are keyword based.

3.1.4. Experiment 1. We have seen that query sizes range from 100 bytes to 30 Kbytes (not considering full image queries which will be studied later where they will be treated as data transfer of large items).

Extensive experiments have been conducted in the Raid lab at Purdue University to study the communication overheads for data items of these sizes. For further details the reader is referred to [21].

3.1.5. Server side communication costs. Queries originate from the client and are transmitted to the server. If the query is specific and requests the data item by name, then the server returns the data item. If an interactive query process is initiated, then the communication from the server is different from data retrieval communication cost.

An interactive dialog between the client and the server during the query process is begun if the user is unable to formulate the query precisely. A typical example is as follows: a user wants to retrieve all images similar to a sunset image. The system retrieves images having a sunset in thumbnail form. These images might not be exactly what the user query wants, but may be close to what she has in mind. She can select one of the images and ask the system to retrieve images similar to that image. The process continues till the user is satisfied.

The size of a user query during a dialog is similar to that discussed above. The size of the data objects the system returns depends on various factors like size of the screen and the number of matches requested. If thumbnail images were retrieved to display on the user's screen, we can estimate that 20 images can fit on a computer screen. If each thumbnail has a size of approximately 80×80, and each pixel is represented by 24 bits, the total number of bytes in each response by the server will be $20 \times 8 \times 80 \times 3 = 384$ Kbytes. Communication costs will be similar to the costs during data retrieval of this size data and will be studied in a later section.

3.2. The browsing phase

We characterize browsing as a process where the user goes through information without a specific query in mind. Browsing the World Wide Web is an example of browsing information. One might begin at the homepage of a graduate student at Purdue University, move on to the homepage of an association the graduate student is interested in, from there move on to the President of the association who also happens to be interested in Space Science, from there move on to the Mars Pathfinder site and so on. An analogy would be walking along library aisles looking at the books on the shelves, and stopping if something looked interesting. This model can be visualized best in hypertext applications such as the World Wide Web, though it is possible to have such a model while browsing through the Purdue University online library catalog.

In a digital library environment where the data sources are highly distributed and replicated, some optimizations are possible to reduce the communication cost while browsing. If a data object is selected to be retrieved while browsing, and if an *equivalent* data item is present at a *network closer* site to the user, then the equivalent data item can be retrieved instead [1]. We define equivalence by abstracting away from the physical details of representation and focusing on the information content of the data item. Some examples of equivalent data objects are:

- Two images of eclipses are equivalent if the purpose of the user is to just clarify some questions she has after reading an article on eclipses
- A photograph of the United States President taken last year and three years before that

This approach has some points of similarity with data replication in distributed database systems. The difference is that instead of explictly replicating the information at several sites, we make use of *existing* similarities among data items and treat them as replicas of each other. Similarity between data items is determined by distance functions. Communication cost can be incorporated into them so that the user can trade off the accuracy of the replica with the reduction in communication cost. Communication cost is measured by *hop count* and predetermined *response time*.

3.3. The data retrieval phase

After querying and browsing users of a digital library application will eventually arrive at a stage when they need to retrieve the actual data item. The path they used could be querying or browsing or a simple data request. The data item is retrieved from the remote site transmitted across the network and displayed on the user's screen or played on the speaker, depending on the media. The user will desire this process to be as fast as possible. We define *response time* to be the delay between the time the user requests the data item and the time it appears on the user's local machine.

As we have observed above, the size of digital library data items are typically large, which is one of the main reasons for the large response time. However, the nature of the media of digital library data objects enables us to perform operations on the data to reduce the response time. One example of such an operation is compression. Another example is reducing the size of the data object by losing some data without making an observable difference to the data object. We will first present the communication costs for large data items. We will then illustrate the possible operations we can perform on each individual media and the reduction in communication cost as a result of these operations.

3.3.1. Experiment 2.

Problem statement. The purpose of this experiment is to measure the performance of communicating 100 Kbytes to 500 Kbytes (example digital library data during the data retrieval phase) in a local area network (LAN), a metropolitan network (MAN) and a wide area network (WAN) [3].

Input parameters. We performed these experiments for data sizes ranging from the order of 100 Kbytes to 500 Kbytes.

Procedure. As in experiment 1, our experiments over the local and metropolitan area networks were conducted between two Sun Sparc workstations **raid11** (Sparc 1) and **pirx** (Sparc 10) in the laboratory and **atom**, a machine in the engineering network at Purdue. Raid11 was the machine used to conduct the experiments and pirx and atom were used as remote sites. The number of hops between raid11 and pirx is one, and they are connected by a 10Mbps Ethernet. The number of hops between raid11 and atom is four. The remote sites in the experiments over a wide area network (WAN) were:

- *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

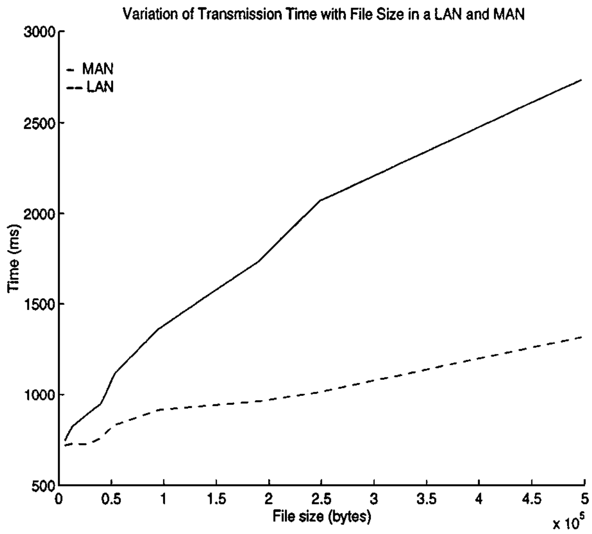


Figure 1. Variation of transmission time with file size in a LAN and MAN.

- *Ironweed.cs.uiuc.edu:* (Illinois). Number of hops = 19
- *Merope.cs.buffalo.edu:* (New York). Number of hops = 19

Results. Figure 1 illustrates the round trip times in a LAN and MAN. The data items under observation range from 6 Kbytes to 496 Kbytes. In a LAN, the round trip times range from 722.84 ms to 1316.82 ms. In a MAN, the round trip times range from 749.41 ms to 2738.63 ms. We can make two observations here. The difference between a LAN and a MAN for a file size of 6 Kbytes is only 26.57 ms. On the other hand, the difference in round trip times for file size 496 Kbytes is 1421.81 ms. The second observation is that the difference in round trip times in a LAN environment between files of sizes 6 Kbytes and 496 Kbytes is only 593.98 ms. The same difference in a MAN environment is 1989.22 ms.

Figure 2 illustrates the round trip times in a WAN. When compared to LAN and MAN the round trip times rise sharply as file size increases and as the number of the hops increases. The difference between the largest size and the smallest size is as high as 23811.691 ms.

Discussion. In a MAN, the difference in communication time between the large file and the small file in our sample is higher than the corresponding difference in a LAN but not as significant as the difference in a WAN. This indicates that in a LAN and maybe even a MAN, large files can be retrieved without any operations such as compression or resolution reduction being performed on them. However, in a WAN the size of the data has to be reduced since there is a significant gain.

3.3.2. Reducing communication time. The above experiments give us a measure of the communication overheads involved while transmitting large data items. We now study each data medium individually and present possible methods for reducing these overheads.

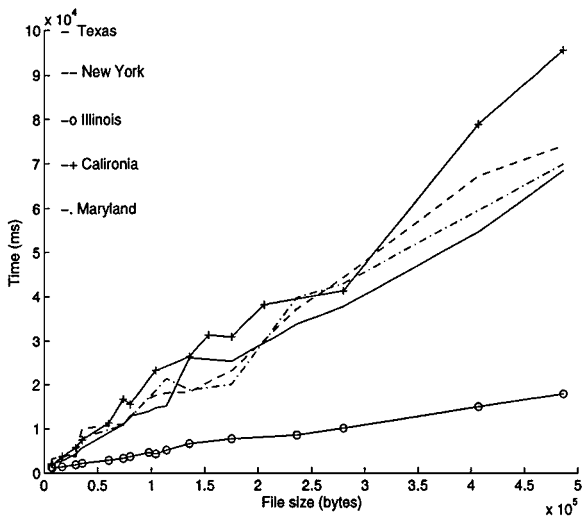


Figure 2. Variation of transmission time with file size in a WAN.

The total communication time consists of the processing time at the sender's site, the transmission time and the processing time the receiver's site. The processing time at the sender's site includes the setup time for the communication protocol used and data processing time such as encoding and compression time. The processing time at the receiver's site includes error checking, arranging data in sequence and data processing time such as decoding and decompression time. Thus, total communication time is:

$$\textit{communication time} = t_{\mathrm{s}} + t_{\mathrm{t}} + t_{\mathrm{r}}$$

where

- t_{s}: Processing time at the sender's site
- t_{t}: Transmission time over the network
- t_{r}: Processing time at the receiver's site

3.3.3. Text data retrieval. Text data transmission has to be lossless. Every character is important and random loss of some bytes will result in messages which appear scrambled. Lossless compression techniques will result in compression ratios of 2 : 1 which will reduce the transmission time by 50%.

Digital library applications can use semantic knowledge of the data to reduce communication time. For example, a document can be represented hierarchically in varying degrees of detail. As shown in figure 3, a body of text can be stored as a document containing abstract and section headings and as another document containing two or three lines of the summary of content [4]. If the user has specified to the system the level of detail which

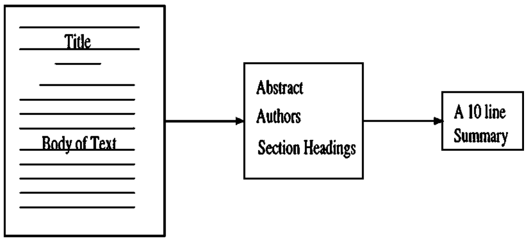


Figure 3. Illustration of decreasing order of detail in hierarchical storage of text.

is sufficient for her, the system can choose the representation which will reduce communication time and yet satisfy the application. Such a scheme requires that the documents be structured accordingly at the sender's site. The structuring can be either manual, or by using automated document processing techniques.

3.3.4. Image data retrieval. Visual data can be stored, viewed, and manipulated in multiple resolutions. Image and video data are rich in semantic content and careful manipulation of the image will result in an image so that no visible information is lost. Thus size of the data can be reduced without lowering information content. For example, lossy compression compression techniques such as JPEG reduce the number of bits used for chrominance and retain the number of bits used for luminance, since the human eye is more sensitive to luminance than chrominance. "Losing" some of the chrominance information does not result in any visible semantic loss in the image but will result in a smaller size file which will be easier to manipulate and transmit. Such a lowering of image quality is acceptable for digital library applications where data is being often retrieved by users who are not experts in the field.

Reducing communication time by trading off quality of the visual data can be achieved in two ways:

1. Lossy Compression of Data
2. Lossy Transmission of Data

Lossy compression of data

There are two basic types of compression: *lossless* and *lossy*. In lossless compression, the original image can be perfectly recovered from the compressed representation, but the compression ratios rarely cross 4 : 1 [10]. Lossy compression techniques attempt to remove redundant information and the original pixel intensities cannot be perfectly recovered. Typical compression ratios range from 4 : 1 to 32 : 1 [10]. The compression ratio depends on the amount of data "lost". The high compression ratio of lossy compression leads to a significant reduction in transmission time. But the compression and decompression results in an increase of processing time at the sender, t_s.

In order to study the overheads involved in compressing data and transmitting it we conducted a series of experiments to measure compression time, communication time, and decompression time. We experimented with compressed visual data using JPEG as a case study. JPEG, a standardized image compression mechanism, is a lossy compression technique [10, 15] which does not reconstruct the original image bit-for-bit but reconstructs an image which looks very similar to the human eye. JPEG primarily stores information on color changes, particularly variations in brightness, because the eye is very sensitive to them [10]. One can choose the extent of compression while using JPEG—10%, 30%, 50%, 75% etc. 10% compression means that approximately 10% of the original file is retained and the decompressed image is reconstructed using this information.

Experiment 4.

Problem statement. To investigate the reduction in transmission time. To study whether increasing t_s and t_r reduces t_t.

Input parameters. 20 images from NASA's repository.

Procedure. We used the free JPEG software developed and distributed by Independent JPEG Group (IJG), which conforms to the ISO JPEG standard. We compressed and uncompressed images and measured the execution times for the processes. The machine used was a Sparc 10 with four 50 MHz processors. Transmission times were measured as before.

Results. Figure 4 shows the response times for a LAN site, MAN site and three WAN sites. The file size used was 400K. Only the three WAN sites have a higher response time for the uncompressed files than the 10% files. Figure 5 illustrates the response times for

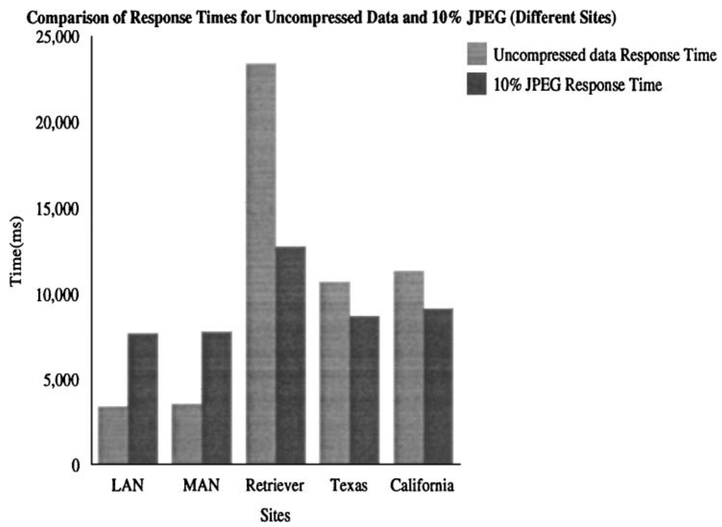


Figure 4. Transmission of 10% JPEG files is viable for large network distances.

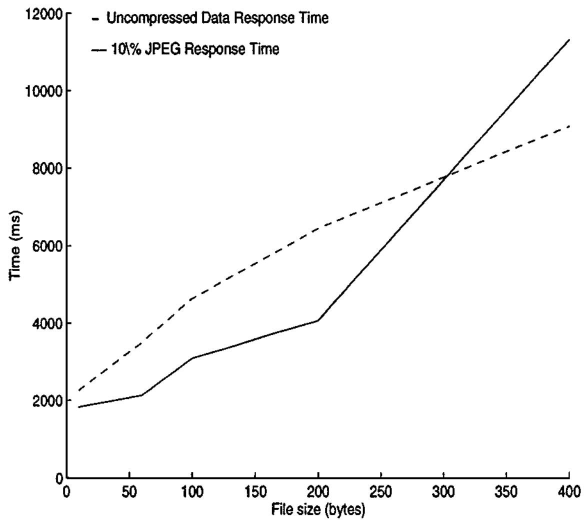


Figure 5. Transmission of 10% JPEG files is viable for large file sizes.

uncompressed and 10% compressed files for a site on a WAN (the file sizes on the X axis refer to the uncompressed file sizes).

Discussion. When data is retrieved over a LAN or a MAN the difference between transmission time of a large file and a small (compressed) file is small and not sufficient to offset the expense of the compression and decompression times. Thus for short network distances the time spent in compression and decompression is more than the time saved in transmission. As can be seen in figure 4, compression, transmission and decompression are only worthwhile in WAN sites. Even for large network distances, if the data size is very small, the saving in transmission time is offset by the cost of compression and decompression as seen in figure 5. This indicates that if the data size is less than a certain threshold, the image can be retrieved as is whatever the network distance.

Lossy transmission of data

We have been assuming so far that reliable transmission has been used. Transmission Control Protocol (TCP) is a reliable transport protocol used by applications that involve data transfer such as FTP and telnet. But at the lowest level, communication is unreliable [6]. Data can be lost or destroyed because of transmission errors, network hardware failure, and congestion. Reliability is achieved by using *positive acknowledgement and retransmission* where it is guaranteed that every packet sent is received by the user. This reliability is achieved at a cost of performance.

As discussed before, visual data is different from traditional text data such as airline information, stock information, patient records, weather information etc. If some text data is lost it cannot be reconstructed from the data that is received. Visual data can be reconstructed from the data received so that the reconstructed data is a good approximation of the original. This is partly because the reconstruction can be effected in a manner such that the errors cannot be perceived by the human eye due to its inherent limitations.

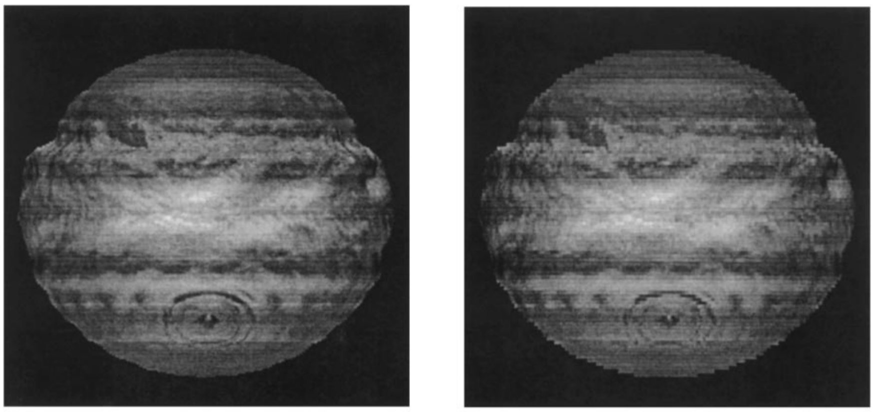
If the reliability requirement can be relaxed for visual data, *best effort* protocols such as UDP (User Datagram Protocol) can be used. Their performance overheads are lowered resulting in a reduction in communication time. The data lost in transmission is reconstructed using the data that is received. The quality of the reconstruction depends on the method used and the amount of data received.

When the data is uncompressed, reconstruction of the image is fairly simple. The basic principle followed by algorithms is to reconstruct the lost pixel using the pixels which are around it. The algorithms vary with respect to which neighboring pixels are chosen to reconstruct the lost pixel.

Packetization of the image

One of the factors the quality of reconstruction depends on is the packetization of the data. The data should be divided into packets such that the loss of a packet causes least distortion to the reconstructed image. For instance, if each row forms a packet, it can be shown that the reconstructed image is very similar to the original image (see figure 6(b) the image has lost 2 out of every three rows and still looks almost the same as (a)). There is another factor which has to be taken into consideration for packetization. The efficiency of the network protocol influences the packet size. Researchers have worked on experimentally evaluating the optimal packet size for a given network protocol [13, 18]. Current UDP implementations use 8 Kbytes as the size of the packet.

When small packets are used, the loss of a packet results in a low percentage of information loss. However, small packets might result in the overhead of header information for each packet.



(a) (b)

Figure 6. (a) Original image and (b) Image with 66.6% loss.

When the data is compressed, reconstruction is not this simple. This is because compressed images contain *control information* such as color tables, quantization tables, and transformation coefficients which is required for decompression. If these data bytes containing the control information are lost the image cannot be decompressed accurately. Thus the effective loss of data is more than the actual loss of data. There are two ways of getting around this. The header information has to be transmitted using a reliable protocol, or each packet has to be compressed individually so that if a packet is lost, it can just be dropped and interpolated from the other packets. The latter approach incurs the overhead of adding the compression header to each packet.

Compressing each packet individually

The image can be divided into packets and compressed individually. If a packet is lost, it does not influence the decompression of any other packet, since each packet was compressed independent of the other packets. Thus the actual loss is the same as the effective loss. But the disadvantage is that each data packet has the overhead of the compression information and the compression ratio becomes low especially for dictionary based schemes since it is difficult to find patterns among small segments of data. Again, smaller packets imply smaller consecutive loss but the total overhead will be high. The disadvantage with transform-based schemes is that some require a minimum number of pixels (JPEG requires 8×8) which places a lower bound on the number of rows that have to be in one packet.

A GIF file contains a header segment, a logical screen descriptor, a global color table and the data. The global color table is optional and can be substituted by several local color tables in the data. The header and the logical screen descriptor are together 17 bytes. The global color table uses three bytes for each color entry and can have upto 256 entries and each entry is a three-byte triplet. Thus the total number of bytes required for the control information is 785 bytes.

A JPEG file contains 20 bytes of header information with an optional extension containing 10 bytes. The quantization tables also have to be included along with the compressed data.

Example. Let us consider an image of size 100 Kbytes. Table 2 compares the difference between compressing the entire image and individual packets of the image. CR is the compression ratio. We observe the difference in compression ratio is negligible for JPEG but 9% for GIF images.

Transmitting control information using reliable protocols

The other approach to transmitting visual data using lossy methods is to transmit the control information using a reliable protocol such as TCP and the data bytes using a protocol without

Table 2. Compression ratio for complete vs. packetized image.

Compression	CR for a complete image	CR for a packetized image
GIF	16.02%	25.82%
JPEG	4.92%	4.93%

the expensive acknowledgement and retransmission mechanism such as UDP. The idea can be extended to transmitting "important" sections of the image using reliable protocols. A judicious mix of different protocols will result in an efficient means of transmitting visual data with very low noticeable loss. We are currently working on experiments to determine the advantages and overheads of such a scheme for various input images.

Quality of visual data

We have said that visual data can tolerate loss and utilized this feature to reduce communication time by using lossy compression and lossy transmission of data. One question that has to be answered is *how much* loss can be tolerated by an application. The related questions are how does one decide that no "visible" information has been lost? How does one measure the information that is lost? We have attempted to quantify the amount of information that is lost in an image or video frame [2]. This quantification will enable us to define quality levels for an image and determine the best quality image or video frame that can be transmitted using the available resources.

We quantify the data lost with respect to the human eye. Different images can tolerate different amounts of loss. We classify images according to the amount of data they can lose. We define three quality levels:

1. Q1: Perfect, no change from the original: This quality is useful for medical imaging applications and scientists and engineers who will conduct further automated processing on the data received.
2. Q2: Indistinguishable from the original with respect to the human eye: This quality is useful when the image will be looked at for only a few minutes. Ex: school children in a virtual classroom session, users browsing through a data repository.
3. Q3: Blurred but user can identify objects: This quality is useful for quick browsing or low bandwidth applications. The loss is noticeable to the human eye but the contents and information contained in the image are clearly perceivable.

We have developed a technique based on the color content of the image to evaluate the quality of the image. A color histogram of an image is a series of bins representing the different colors in the image. Each bin contains the number of pixels in the image that have that color. The advantage of using color histograms is that they are invariant to slight modifications of position and scaling and hence provide a more accurate measure than pixel-to-pixel matching.

Our technique takes as input the histograms of the original image and the lossy version of image and outputs a value *col_diff* which is a measure of the color difference between the two versions of the image. The function used to calculate the difference is a variation of the Euclidean measure. *col_diff* is compared with a specified threshold. The threshold depends on the quality level required by the user. If *col_diff* is less than the threshold then the quality of the image is acceptable to the user.

Our technique works well while evaluating whether an image is *indistinguishable* to the human eye when compared to the original. We have experimentally evaluated the technique and compared it with the pixel-to-pixel matching technique. Our results show that our technique is more accurate in evaluating loss with respect to the human eye. For further details the reader is referred to [2].

3.3.5. Video data retrieval. Video data can be treated as a stream of images. The techniques described above for the efficient transmission of images apply to video data. Since video data is continuous, there are some issues specific to video data which are addressed in this section. The approach developed in our laboratory is based on dynamic adaptability of the quality of video transmission to the bandwidth conditions.

Adaptable transmission of video data. Video transmission applications have to maintain a constant frame rate. The current TV frame rate is about 30 frames per second. The variation in available bandwidth does not allow this frame rate to be maintained without reducing the amount of data by trading off some aspects of video quality. The quality of the video has to be dynamically changed to maintain the frame rate of 30 frames per second. We have identified four aspects of video quality that can be changed to adjust to the available bandwidth:

- Color depth compression: Color video can be compared to gray-scale video to reduce the size of the data since gray-scale pixels require fewer pixels to encode than color pixels.
- Frame resolution reduction: Replacing every 2×2 matrix of pixels by one pixel reduces the size of the video frame by a factor of 4. The image is reconstructed at the receiver to keep the physical size of the frame unchanged. Since the resolution reduction process is lossy, the receiver gets a frame which is an approximation of the original.
- Frame resizing: The frame size is changed to reduce the size of the data. For instance, reducing the frame size from 640×480 to 320×240 reduces the bandwidth requirement to 25% of the original.
- Codec schemes: Different coding schemes have different compression ratios. Typically, schemes with high compression ratios require more time to compress but the smaller compressed frames can be transmitted more quickly. If the bandwidth available is extremely limited, it might be worth while to reduce the communication time at the cost of computation (during compression) time.

Our research group has conducted several detailed experiments to test the feasibility of the above ideas and have come up with a framework to determine the parameters of quality that should be used for video transmission. The framework allows the quality of video transmission to adapt according to the bandwidth available. For further details the reader is referred to [12].

3.3.6. Audio data retrieval. As with image data, audio data can be compressed and transmitted using both lossy and lossless compression schemes. However, unlike image and video data, audio data can tolerate less loss. One might watch a movie with a blurry screen but would not listen to music on a radio with a lot of disturbance. The loss tolerated can also vary with the content of the audio data. More loss can be tolerated in a speech than in music.

Disturbance in a President's speech might be acceptable when compared to Beethoven's ninth Symphony.

We are currently conducting experiments with transmitting compressed audio data using both lossless and lossy transmission mechanisms. We are also investigating adaptability schemes similar to that of video data.

3.3.7. The electronic commerce model. Digital library applications might have an environment similar to that of electronic commerce when there is a cost associated with the data and the information provided. Examples of such applications are lending libraries which charge on a per day basis, libraries which charge fine for a late return, libraries which charge for downloading documents on a per document basis. Supporting payment by a client brings in 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 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 an overhead to the data retrieval process.

4. Continuous media

Continuous media. We define a continuous media data item as a data item containing a series of data points which form a sequence with respect to time. Video and audio are continuous data items. A continuous media data item can be viewed as a series of discrete data items. Video consists a series of images but there are some special issues involved in transmitting continuous media data items which are not applicable when we are dealing with only discrete data items. A user query requesting a continuous data item will consist of:

- Transmission of query.
- Browsing: There are two modes of display when a user is browsing a respository of continuous data. Clippings of the actual data item can be displayed or played (depending on whether the data item is audio or video). The data items would have to be played or displayed sequentially. Some users might find it time consuming and prefer an alternate mode of display, which is to use still media data items to represent the continuous media data items. Video clippings can be represented by single frames and/or text abstracts and audio clippings can be represented by text abstracts.
- Retrieval of data: Once the presentation of the continuous media data item has begun, a guaranteed rate has to be sustained.

The communication overheads for transmission of a query and the browsing mode are similar to those discussed in previous sections. The data transfer will be of the order of a few kilobytes. Here we will discuss the issues involved in the transfer of the entire continuous media data item.

4.1. Modes of transmission of continuous media data

There are two ways to transmit continuous media data items:

4.1.1. Store and display. In this mode, the entire data object is received before presentation begins. The initial startup time latency is high, but once the presentation begins, it proceeds smoothly and at the required frame rate. The initial latency will include the time to receive the data object and the time to decode it (the decoding time for the entire file will be larger than for smaller pieces of the data). The biggest advantage is that the frame rate can be maintained with no overhead once the entire data object has been retrieved. The disadvantages are that this mode is not feasible in real-time applications and the initial startup latency can be very high, often the order of 10 minutes for even a 2 minute video or audio data object. The start-up latency can be particularly annoying to the user if she decides that that is not what she wants after a few seconds of viewing.

4.1.2. Continuous. The display or playing of the data item proceeds concurrently with retrieval. Data is displayed as soon as the amount of data sufficient for the first presentation and atleast equal to the latency of fetching the second piece of data is received. When the first piece of data is being displayed, the second piece of data is retrieved concurrently. There are several advantages to such a scheme:

- We need to maintain only a reasonably large buffer space, not as large as the entire data file.
- We can monitor the underlying network and adapt to changes by dynamically adjusting buffer size at the client.

The disadvantage is that there is a possibility of the pieces of data not arriving on time resulting in a presentation which is not smooth. The requirements on the system are as follows:

- Reduce lag as much as possible
- Enough buffer space to sustain display of playing of data while the next chunk of data is received and decompressed
- High computing power to have fast compression and decompression

5. Overheads while transmitting multiple media

As we discussed in the introduction of this paper, digital library documents are multimedia documents and can contain text, image, video, audio in a single organized package. The communication overhead while transmitting a document containing data from multiple media is the equal to the overhead of the component of the document which has the maximum overhead. This is because the retrieval of all the components of the document is begun simultaneously.

Synchronization is one of the issues that has to be dealt with while transmitting multiple media together. The most common combination is the transmitting of audio and video data. We have not dealt with issues related to synchronization in this work.

6. Communication bottlenecks

We have been using the terms 'communication bottlenecks' and 'communication overheads' fairly frequently. In this section we will identify the different factors which contribute to the communication overheads. The categorization and understanding of the different factors is useful in deciding the applicability solutions to overcome the bottleneck.

Communication delays are caused by the following factors:

- Physical capacity limitation of the communication link: The different physical media have an inherent capacity limitation.
- Technology: Some of the currently available network technologies are circuit switching, packet switching, frame relay, cell relay, FDDI, and ATM). Some of the technology specific factors which influence communication delay are: cell setup delay, packet formation time, total lost frames/packets/cells, load balancing/load sharing limitations, masking, filtering, and forwarding rates, error detection and correction efforts, and level of redundancy.
- Number of hops: The number of hops between the sender and receiver gives a rough estimate of the network distance between them. Every time the data is forwarded by a router, it is referred to as a hop. At each hop there is delay due to speed of hardware and software interfaces, memory and buffers, address database look-up, address verification, processing, filtering and forwarding of packets, frames, and cells.
- Traffic: The physical capacity which is bounded has to be shared among different applications. The bandwidth allocation scheme determines the network bandwidth allocated to a given application. There are several bandwidth allocation schemes and over a public network such as the Internet, they follow a 'fair' policy which ensures that no application is deprived a share of the network bandwidth. Consequently, the network bandwidth available for existing applications is reduced when a new application requests bandwidth.
- Buffer limitations: The buffer limitations at the nodes at either end of a communication path and the routers on the communication path also contribute to the communication delay. The buffer might not be able to store all the packets which arrive and hence some packets are dropped. This results in re-transmission (in a lossless protocol such as TCP) and consequently more contention for existing network bandwidth.
- Out-of-sync CPU: CPU speed is slower than network speed. Packet or frame or cell processing functions such as packet formation, address lookup, instruction execution, buffer filling time and error checking have their speed bounded by the computation power of the CPU.

7. Defining quality of service parameters for digital libraries

Quality of service (QOS) specifications are used by distributed multimedia systems to enable applications and users request a desired level of service. The system attempts to satisfy the specifications, and if that is not possible due to resource availability restrictions, the application can enter into a negotiation with the system. During the negotiation process the QOS specifications are changed so that the system can meet the requirements. The process might continue over several iterations.

An example of QOS parameters can be found in video-conferencing applications. This is a real-time application and needs a guaranteed supply of system resources to sustain a uniform level of performance. Some of the parameters are loss rate, throughput, frame rate, response time and presentation quality. We propose that similar QOS parameters be associated with a digital library system. This will allow the application or the user to negotiate with the system and arrive at a set of values for the parameters which both satisfy the user and can be supported by the system. The application can trade-off some parameters in exchange for others.

- Accuracy of information: Digital library queries are satisfied not by exact matches as in traditional databases, but by similarity matches. The accuracy of the match between the query and the retrieved data item can be specified as a QOS parameter. For example, consider a query requesting the 1997 annual report for the Computer Science department. A 1995 annual report is a match, but not as accurate as the 1997 annual report. Depending on the requirements of the application, the lower accuracy match might satisfy the user. (For instance if she needed to find out the graduate student intake every year then it doesn't matter which annual report she uses. On the other hand, if she wanted to know who the head of the department was, then there is no guarantee that the data is accurate in the 1995 annual report. The head could have changed.)

  Lower accuracy of match will result in a higher availability. This is because a lower accuracy implies that more data items match the query. Thus, there is a higher probability of finding a data item which is available and at a site network close to the user.
- Precision of information: Precision is a concept borrowed from information retrieval. It is defined as:

$$precision = \frac{Number\ of\ relevant\ documents\ returned}{Number\ of\ documents\ returned}$$

  Relevance of a result can be in the context of level of accuracy described above. To achieve a high precision, more computation is required to eliminate the data items which do not belong to the answer set.
- Recall of information: Recall is a related concept which is also borrowed from information retrieval. Recall is defined as:

$$recall = \frac{Number\ of\ relevant\ documents\ returned}{Total\ number\ of\ relevant\ documents\ in\ the\ collection}$$

As with precision, relevance of the result can be in the context of the level of accuracy. Here too, more computation is required to achieve a high percentage of recall. All the data items have to be accessed to ensure that no matches are missed. Along with a higher computational cost, this could result in a higher communication cost since *all* the data items have to be accessed.

- Data comprehensiveness: Some components of the item can be specified as not required to reduce system and communication requirements and response time. For instance, a video clipping can be retrieved without the audio component. A text document can be retrieved without the images. Components with a higher computational and communication overhead such as multimedia data can be ignored if they are not required to satisfy the query.
- Presentation quality: Data items can be presented at different levels of quality. Visual data items (images and video) can be presented at different resolutions, different color depths, different sizes, and different codec schemes. Audio can be presented at different sampling rates. The presentation quality can be a user specified parameter. An application might prefer a lower quality data item if the response time can be lowered. For example, a K-12 student would be satisfied with a lower resolution medical image than a medical student.
- Response time: We define response time as the time between the instant the user submits a data retrieval request and the time the data appears on the screen. This can be several seconds or even minutes. Several applications would like tradeoff the quality of the data, accuracy of data, precision, and recall in exchange for a lower response time.

The application or user can specify the different parameters desired. Upper and lower bounds can be used to express acceptable situations. From a communication point of view, the goal is to minimize response time and maximize accuracy of information, precision and recall of the data retrieved, presentation quality and comprehensiveness of the data. Relaxing the constraint on one of the parameters which has to be maximised, might help in maximising the other parameters.

8. A communication framework for digital libraries

We have now seen a comprehensive and detailed description of the communication overheads and possible solutions in digital library operations. We have seen that a digital library operation consists of several smaller problems, and each has its own solution. Our goal was to develop a infrastructure that supports the interactions of the different solutions with each other. In this section we present the communication infrastructure which addresses the relevent issues and incorporates the solutions described above to provide efficient communication support for digital library applications.

Our digital library data model consists of a client which sends the user requests in the form of queries and a server which services the user requests. The framework includes communication components at both the server and the client. Figure 7 presents the architecture.

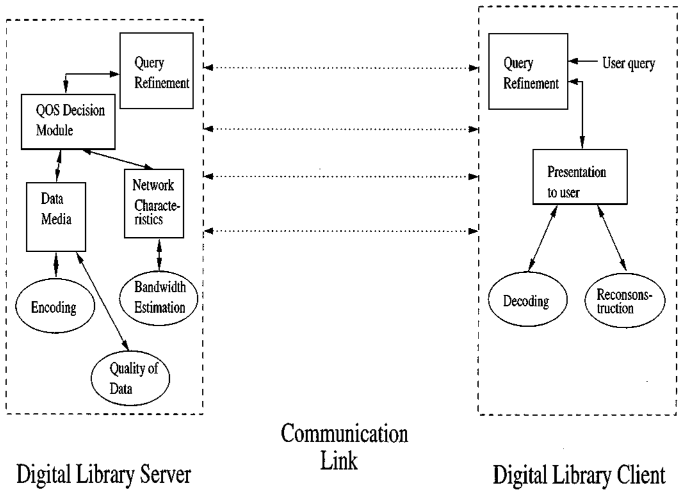


Figure 7. Communication framework for digital library applications.

8.1. The digital library server

At the server, the components are decision modules, utility functions, and the communication interface with the client.

8.1.1. Decision modules. The decision modules take some parameters as input and outputs the steps that the server should follow. The different decision modules are designed to address the different modes of digital library operations.

Query refinement module: This module interacts with the client when an interactive query mode is required. It interacts with the data repositories to retrieve data to assist in the interactive process and for browsing. If an interactive mode is not required it simply passes on the query to the QOS decision module.

Network characteristics module: This module takes as input the current network characteristics such as bandwidth available, network distance of the client, and reliability of the network. It then makes decisions as to the parameters for the data transfer. Low bandwidth availability and high network distance will indicate that the data size should be reduced as much as possible maybe using lossy compression techniques. Low reliability will influence the encoding method since some data packets might be lost during transmission.

Data media module: We have seen that the communication parameters vary according to the media of the data item. This module determines the media and provides input to the network characteristics module to help it determine the means for efficient data transfer.

QOS module: This module takes as input the QOS parameters specified by the client. It then verifies whether the system can provide the required service. This verification process is conducted in consultation with the network characteristics module and data media module. If the user specifications cannot be met, it negotiates with the client till an acceptable specification of QOS parameters is reached. This negotiation is conducted through the query refinement module so that user can use the same user interface as for the interactive querying.

8.1.2. Utility functions. The decision modules call utility functions to obtain characteristics which will help them in the decision making process. These utility functions are:

Bandwidth estimation: This function estimates the bandwidth available by sensing the traffic between the server and the client.

Encoding: This function performs encoding of data and also provides an estimate of the resources (time and system resources) necessary for the encoding. This function is used at the server side.

Quality of data: The data media module uses this function to compute an acceptable level of quality for the data depending on the user specifications. The function chooses a method depending on the media: naturally, audio, image, and video have different quality control measures.

8.2. Communication link

The communication link between the server and the client should support various network technologies and protocols. The network technologies we envisage are IP and ATM and the protocols that we envisage are TCP, UDP, and RTP.

TCP (Transmission Control Protocol) provides reliable stream delivery. It can be used for transmission of messages which cannot afford loss such as the user queries and text data. UDP (User Datagram Protocol) is a *best effort* protocol. Reliability cannot be guaranteed, but *bound* on the loss could be guaranteed [19]. This protocol could be used for bulk data transfer of multimedia data when some loss can be tolerated. RTP (Realtime Transfer Protocol) is a protocol developed for video transmission and can be used for continuous media digital library data transmissions.

8.3. The digital library client

The digital library client architecture is less complex than that of the server.

8.3.1. Decision modules at the client.

Query refinement module: The functionality of this module is the same as the query refinement module in the server. It displays the messages it receives from the server to the user and reformulates the query based on the user's refinement.

Presentation to the user: The data received from the server could be in one of three forms: it could be data which can be presented without any further processing to the user, it might be encoded data which has to be decoded and it might be data transmitted using a lossy transmission protocol and hence require reconstruction at the client's site. This module determines the kind of data that it receives and interacts with the utility functions to obtain the data in the final presentation form.

8.3.2. Utility functions at the client.

Decoding: This function decodes the data, after determining the media type and the encoding scheme used.

Reconstruction: If lossy transmission techniques are used to transmit data, this function measures the loss, if any, and reconstructs the lost packets of data using an interpolation function which depends on the media type, quality of data required by the user and response time acceptable to the user. This function also reconstructs data if the data is intentionally scrambled at the server's site to avoid loss of continuous data bytes.

9. Digital library services in a mobile environment

Digital library users include not just users on a fixed local area network or a wide area network, but users with mobile computers and wireless links too [4]. Provision to access digital library services through wireless networks is required by a wide range of applications from personal to research to customized business computing. For instance, archeologists working on remote locations may need access to library data related to their discoveries. Travellers passing a signboard on a highway saying 'next exit 25 miles' would like to know the restaurents within a five mile radius of that exit.

Mobile computing environments are characterized by frequent disconnections, limited computing power, memory, and screen size of the mobile hosts, and varying modes of connection: fully connected, partly connected, and doze mode for conserving energy.

Mobile computing environments have to address some special issues while providing digital library services when compared to wired environments. They include:

- Disconnected operation: Users of digital library services will not be continuously connected to the server providing the information.
- Weak connectivity: Wireless networks deliver much lower bandwidth than wired networks and have higher error rates [8]. While wired network bandwidth of 155Mpbs has been achieved, wireless communication have only 2Mpbs for radio communication, 9–14Kpbs for cellular telephony [8], and 250Kpbs–2Mpbs for wireless LANs.
- Asymmetric capabilities of server and client: Most of the static servers have powerful broadcast transmitters while mobile clients have little transmission capability.
- Varying client location.
- Variant connectivity.

The above characteristics result in the following features a digital library in a mobile environment should contain:

- Broadcast: Some information in digital library applications are very relevant in a broadcast scenario. Some examples are traffic patterns, weather information, travel information (motels, restaurents etc.), and other public services. The server has powerful transmission capabilities and computing power when compared to the client making broadcast one mode of communication of information. The mobile clients need not be connected all the time. They can be in doze mode and only connect to the network when they need information. When they are connected they can direct queries to the server regarding the information that is broadcast and perform further processing on the information picked up from the broadcast before presenting it to the user.
- Batch processing of queries: The asymmetric computing capabilities of the static server and the mobile client coupled with the weak connectivity make interactive query processing inefficient. To accomodate the disparate capabilities, the static server should be able to execute a query with as little interaction as possible with the mobile client, that is, the mode of operation should be batch rather than interactive. In a batch mode, the mobile client can submit a query, voluntarily disconnect to conserve power, and then reconnect to obtain the result of the query. The amount of the processing on the static server should be maximized and the amount on the mobile client minimized. To achieve this, we borrow the concept of the *user's profile* from information retrieval. Information about user's interest and recently asked queries are stored. Input from the user's profile is used by the static server to resolve ambiguities in the query and without interaction with the mobile client. Such an approach is feasible for specific domains.
- Delayed response queries: When queries are not time critical, mobile users can submit the query and disconnect and pick up the query either at the base station where the query was submitted, or at the next base station. The static server will execute the query and save the result and the user can access it whenever she wants to.
- Location parameters: Location information has to be included in mobile queries. A query can be interpreted differently based on the location of the mobile client. Location has both physical and temporal coordinates. An example of query which will use physical coordinates is "retrieve all restaurents within a 5 mile radius of this car." An example of a query which will use temporal coordinates is "what is the taxi availability of the location I am in?".

A layered architecture for providing digital library services to mobile users has been developed at the Raidlab at Purdue University. The architecture addresses the issues we have discussed above and incorporates the features we have described. For more details the reader is referred to [4].

10. Impact on digital libraries if communication is free

Our experiments have shown that communication costs dominate the response time in a wide area network. Algorithms for implementing a database in a distributed environment have been designed with this view. What would happen to digital library databases if communication delay was negligible or free?

- Replication: Traditionally, replication algorithms have sought an optimal placement so that communication cost is minimized. They often provide local copies. Maintaining consistency among a large number of copies is expensive. But when communication is free, that would not be the case since messages can be sent instantaneously. Also, there is no need for a local copy. Replication would be just to provide availability in the case of failure. If there are fewer copies the number of accesses can be a bottleneck.
- User interface: When communication is free, users will tend to get more data than their precise need. The functionality of the user interface will increase in terms of presenting the data and helping the user browse, navigate, and search the data retrieved.
- Content-based retrieval: When user queries are based on content they may not be precise. Query processing will be an iterative process with the user refining his query at each step based on system output. We have seen that such a process can be communication intensive. When communication is free, this has no effect on response time. This gives the users more flexibility while searching.
- Indexing and searching: Now the time spent for indexing and searching is a small percentage of the response time in a distributed system because of the dominance of communication cost. If communication cost is zero, more time and effort should be spent in developing complex searching algorithms since time for these algorithms will dominate.
- Parallel processing: When communication is free, all parallel processing algorithms can be applied to a distributed environment. For example, several complex fractal coding algorithms can be executed using a parallel algorithm in a distributed environment making the use fractal coding viable for indexing images [20].

Acknowledgment

This research is supported by a grant from National Science Foundation under NCR-9405931 and a NASA fellowship.

References

1. M. Annamalai and B. Bhargava, “Defining data equivalence for efficient access of images in a distributed environment,” in Proceedings of the Second World Conference on Integrated Design and Process Technology, Society for Design and Process Science, December 1996, pp. 11–18.
2. M. Annamalai, A. Sundaram, and B. Bhargava, “A color-based technique for measuring visible loss for use in image data communication,” in Lecture Notes in Computer Science Series (LNCS), Vol. 1134, Springer-Verlag, September 1996, pp. 39–48, Database and Expert Systems Applications (DEXA ’96).
3. B. Bhargava and M. Annamalai, “Communication costs in digital library databases,” in Lecture Notes in Computer Science Series (LNCS), Vol. 978, Springer-Verlag, September 1995, pp. 1–13, Database and Expert Systems Applications (DEXA ’95).
4. B. Bhargava, M. Annamalai, and E. Pitoura, “Digital library services for mobile computing,” SIGMOD RECORD, Vol. 24, No. 6, pp. 34–39, 1995.
5. B. Bhargava, S. Li, and J. Huai, “Building high performance communication services for digital libraries,” in Proceedings of the Forum on Research and Technology Advances in Digital Libraries, McLean, Virginia, May 1995.
6. D.E. Comer, Internetworking with TCP/IP, Prentice Hall, 1995, Vol. 1, p. 569.

7. R. Entlich, L. Garson, M. Lesk, L. Normore, J. Olsen, and S. Weibel, "Making a digital library: The chemistry online retrieval experiment," Communications of the ACM, Vol. 38, No. 4, p. 54, 1995.
8. G.H. Forman and J. Zahorjan, "The challenges of mobile computing," IEEE Computer, Vol. 27, No. 6, April 1994.
9. E.A. Fox, R.M. Akscyn, R.K. Furuta, and J.J. Leggett (Eds.), "Special issue on digital libraries," Communications of the ACM, April 1995.
10. R.M. Gray, P.C. Cosman, and E.A. Riskin, "Image compression and tree-structured vector quantization," in Image and Text Compression, Kluwer Academic Publishers, 1992, pp. 3–34.
11. B.G. Haskell, A. Puri, and A.N. Netravali, Digital Video: An Introduction to MPEG-2, Chapman and Hall, 1997.
12. S. Li, "Quality of Service Control for Distributed Multimedia Applications," PhD thesis, Department of Computer Sciences, Purdue University, 1997.
13. X. Liu and B. Bhargava, "Effective strategies for multimedia message transmission: UDP cooling and TCP," Technical Report CSD 95-064, Department of Computer Sciences, Purdue University, October 1995.
14. D. Merrill, N. Parker, F. Gey, and C. Stuber, "The university of california CD-ROM information system," Communications of the ACM, Vol. 38, No. 4, p. 51, April 1995.
15. J.D. Murray and W. Vanryper, Graphics File Formats, O'Reilly and Associates, Inc., 1994.
16. B. Schatz and H. Chen (Eds.), "Special issue on digital libraries," IEEE Computer, May 1996.
17. M. Stonebraker, J. Frew, K. Gardels, and J. Meredith, "The Sequoia 2000 storage benchmark," in Proceedings of the 1993 ACM SIGMOD, ACM, May 1993, pp. 2–11.
18. P. Vaidyanathan and S.F. Midkiff, "Performance evaluation of communication protocols for distributed processing," Computer Communications, Vol. 13, No. 5, pp. 275–282, June 1990.
19. S. Wang and B. Bhargava, "Multi-Pass transmission policy: An effective method of transmitting large multimedia objects in the wide-area network," in Proceedings of the 21st Annual International Computer Software and Applications Conference, Washington, D.C., August 1997, pp. 382–387, IEEE.
20. A. Zhang, B. Cheng, and R. Acharya, "Texture-based image retrieval in image database systems," in Proceedings of the 6th International Workshop on Database and Expert Systems, October 1995.
21. 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, October 1993.

Bharat Bhargava received his B.E. degree from Indiana Institute of Science and M.S. and Ph.D. degrees in EE from Purdue University. He is a professor of computer sciences at Purdue University. Prof. Bhargava's research involves both theoretical and experimental studies in transaction processing in distributed systems. He has conducted experiments in large scale distributed systems, communications, and adaptable video conferencing systems. Currently he is conducting experiments with research issues in large scale communication networks to support emerging applications such as digital library and multi-media databases. He is also investigating security issues in multimedia and mobile systems.

Along with John Riedl he won the best paper award for their joint work on a model for adaptable transaction systems. Prof. Bhargava is on the editorial board of two international journals including Multimedia Tools and Applications. He founded and served as chairman of several ACM and IEEE conferences including the symposium on Reliable Distributed Systems. Prof. Bhargava is a fellow of the Institute of Electrical and Electronics Engineers and the Institute of Electronics and Telecommunications Engineers. He has been awarded the Golden Core charter member distinction by IEEE Computer Society for his distinguished service and received the Outstanding Instructor award from the Purdue chapter of the ACM in 1996 and 1998. In 1999 he received the IEEE Computer Society's Technical Achievement award.

Melliyal Annamalai is a principal member of the technical staff in the interMedia group at Oracle Corporation. She has a Ph.D. in Computer Sciences from Purdue University, where she was a NASA graduate fellow and an M.S. from the University of Maryland, Baltimore County where she also worked at the National Institute of Standards and Technology.

Her research interests include E-commerce multimedia applications, infrastructure for Internet multimedia information systems, and visual information retrieval. Most recently, she has been working on the design of efficient and robust multimedia information systems, with specific emphasis on content-based retrieval of images. Much of her work is analytical, but oriented towards real-world problems.

She is a member of the ACM, ACM SIGMOD, ACM SIGMM, and the UPE. She has been on several program committees and is active in mentoring young computer scientists at the undergraduate and graduate levels.