

# Real Time Medical Image Consultation System Through Internet

D Durga Prasad†, Saikat Ray†, Arun K. Majumdar\*, Jayanta Mukherjee\*,  
Bandana Majumdar†, Soubhik Paul†, Amit Kumar Verma†

*Department of Computer Science and Engineering,  
Indian Institute of Technology, Kharagpur, India*

*\*{akmj,jay}@cse.iitkgp.ernet.in*

*†{ddprasad,skt.saikat,bandana.majumdar,paul.soubhik,amitkumarverma1985}@gmail.com*

## ABSTRACT

Teleconsultation among doctors using a telemedicine system typically involves dealing with and sharing medical images of the patients. This paper describes a software tool written in Java which enables the participating doctors to view medical images such as blood slides, X-Ray, USG, ECG etc. online and even allows them to mark and/or zoom specific areas. It is a multi-party secure image communication system tool that can be used by doctors and medical consultants over the Internet.

**Keywords:** Web Server, Telemedicine, Tele Consultation, Data Conferencing, Graphics Communicator

## 1. INTRODUCTION

Medical imaging plays a very important role in diagnosis of diseases. Teleconsultation between doctors would become more meaningful if a tool is available that not only supports verbal communication but displays medical images at their consoles with enriched features such as marking, text annotating, zooming etc. Implementing such a tool over the Internet is a challenging task as it involves transfer of different types of contents of large volume of data between the clients. Any such exchange is also fraught with the problems of data security. This paper presents a “Real Time Image Consultation System”, named as Online Graphics Communicator (OGC), which uses Internet as the communication infrastructure. OGC proposed in this paper provides an excellent tool for teleconsultation among doctors in a telemedicine system to facilitate discussion over medical images of the patients. We first look into the needs of such a system and its role in the diagnosis of various diseases. A description of the system is then presented high-lighting its different features from a user’s perspective. Finally, the paper concludes with a discussion on the scope of further development.

## 2. BACKGROUND

Although not many systems have been reported in the literature providing aforementioned features over internet, contributions made by a group of researchers in Sydney [1] and in

Taiwan [2,3,4] have similar objectives. The system described in [1] used HTTP protocol to deal with online consultation of a patient's image over the Internet. However, it mentioned the use of JAI<sup>1</sup> that often adds an extra overhead to the client machines. Whether more than one image can be handled simultaneously was also not clearly mentioned.

The system described in [2] used ftp as a means of communication to transfer images of a patient to the client's local machine. It has three disadvantages:

- 1) It used ftp for file uploading and downloading which is often blocked because of security considerations.
- 2) It stored the image at the client site. Hence a suitable client tool needs to be installed for local storage management. In addition, retrieval of files and relevant image data from server as well as its storage on client side involves additional IO operations thereby increasing response time.
- 3) It functioned on a peer to peer basis. Peer-to-peer communication has the disadvantage that most proxies in today's environment do not allow a client to directly connect to the outside world to avoid security breaches.

References [3] and [4] did not address the problems related to usage of the respective telemedicine systems over Internet taking security threats into account. Since most firewalls do not allow any ports other than 80 for HTTP or 443 for HTTPS, the usage of some of these schemes would usually be limited to Intranet or VPN environment, where additional ports such as 21 for ftp may be permitted. There are also other systems that provide online archival, storage and communications. One of them is Picture Archiving and Communication Systems (PACS) [5]. To use such systems one needs to have specific hardware and software installation on their machine. This is often costly and hence limits its accessibility.

### 3. SYSTEM DESCRIPTION

The proposed Online Graphics Communicator (OGC) provides a platform for doctors who are remotely connected and would like to participate in an online session. During such a session, they can open patients' images and perform various operations on it. This is implemented using a Java Applet<sup>2</sup> that executes inside a browser.

#### 3.1 Features

The system proposed in this paper deals with teleconsultation among doctors examining medical images and other information related to a patient. When a doctor refers a case to another doctor for consultation, it would be necessary to share relevant medical images. To undertake such sharing the following options are made available:

- To send images and medical records beforehand to all participating sites where the relevant data are saved.

---

<sup>1</sup><http://java.sun.com/javase/technologies/desktop/media/jai/> , Java Advanced Imaging

<sup>2</sup><http://java.sun.com/applets/>

- To send commands from client sites to fetch relevant image/record during a consultation.

Since both these approaches tend to store the data in a stable storage (disk), additional disk access time would be involved. The proposed system addresses the data fetching and display of data in a slightly different manner. When a user (doctor/consultant) logs into the system, he/she is presented with thumbnails of relevant images. The relevant full image is fetched online, loaded into the memory directly and opened when the thumbnail is clicked. This avoids storage of image/data in the local stable storage. Images are fetched from the server to the client on-demand. Moreover, all images of the patients need not be downloaded to the client beforehand.

In order to protect internal machines and databases from Internet based threats, most organizations use proxies/firewalls that implement Network Address Translation (NAT)<sup>3</sup> and allow only HTTP/HTTPS traffic because of security considerations. In view of this, OGC has been developed and tested in a secured network where only HTTP and HTTPS are allowed and everything else is blocked. It is assumed that images and patient records are maintained at a central webserver and a client sends HTTP requests to the webserver for fetching data. Images and/or messages are received as HTTP responses. The client does not have to install any additional software. OGC makes HTTP requests from the clients to webserver only if permitted by the client. A self signed certificate is presented to the client for verification.

Since OGC has been designed to help doctors who want to consult and discuss with images related to a patient, such images should not only be readable but must be presented in a user friendly manner. OGC enables a doctor to open any image or patient record which is also simultaneously displayed on the other terminals of the consulting parties. If a doctor writes a text on an already opened image, it is simultaneously displayed on other windows of individual machines where this image is being viewed. Besides image chatting, OGC supports text based chatting to start a normal discussion before actually starting a conference related to a particular patient. Some of the important functionalities of OGC are listed below:

1. Open an image online during discussion.
2. Mark on the specified regions of the image.
3. Zoom selected areas in the image.
4. Zoom whole image.
5. Write text on the image.
6. Write text on the zoomed image.
7. Open specialized canvas for displaying skin patches or the different organs of the body.
8. Write text for text-based chatting.
9. Display the doctor's status.

---

<sup>3</sup>RFC3022, Network Address Translation or Network Address Translator, refer to [www.ietf.org/rfc/rfc3022.txt](http://www.ietf.org/rfc/rfc3022.txt) .

10. Perform certain mathematical operation on an image portion (e.g. an ECG image) and display the result to other participating doctors.
11. View any image with any resolution. If the image is too large, it can be scaled down to fit the screen. If the image is too small in resolution, the zooming facility can be used to see an enlarged image.

OGC does not impose any restrictions on the number of users of the system, or on the number of users in a group. The number of concurrent connections would only be limited by the capabilities of the webserver. There is cross platform compatibility as it is developed in Java. Since the communicator uses HTTP requests to send and receive data over the Internet, this mode of communication enables any client to access the server irrespective of their underlying network links such as leased lines, ISDN, DSL, PSTN etc.

### 3.2 System Design and Architecture

OGC is a tool of Telemedicine Server, iMediK<sup>4</sup> [6,7], developed at Indian Institute of Technology, Kharagpur, India. The iMediK was designed for Windows platform. It is a portal built with an intention to provide access to quality healthcare to the remotest regions through innovative and state of the art technologies. The iMediK was developed with an objective of providing a secured telemedicine platform over the Internet. A brief description of this system is provided below.

#### 3.2.1 iMediK – A brief overview

The iMediK is built on a four layered architecture (shown in Figure 1). The different layers of the system are the following [6,7]:

- **Database Layer:** The Clinical Database forms the lowest layer of the application. It contains all the medical records of the patients as well as personal details of the users. The database has different logins for different types of users and the application server connects to the database using the appropriate login for the current role of the user. The database server is accessible only from the application server.
- **Application Layer:** The Application Layer or the Business Logic Layer is the core of the application. It is responsible for all communications with the database. This layer intercepts all data requests from the presentation layer. If the current user is authorized to access the requested data, it fetches the data from the database, runs application logic on the fetched data and returns the final output to the presentation layer. Similarly data insertion and other user requests are also handled by the Application Layer.
- **Presentation Layer:** This layer resides between the Application Layer and the Web Proxy Layer in the demilitarized zone (DMZ). It is the segment of the application which is responsible for organizing the data fetched by the Application Layer into a user friendly format. In the iMediK system, the Presentation Layer formats the data into html format for external communication.

---

<sup>4</sup><http://tmportal.iitkgp.ernet.in>

- **Proxy Layer** [6, 7]: The Web Proxy Layer is the only portion of the software that resides in the DMZ. This is the publicly accessible portion of the application which communicates with the client. Whenever a request comes from a client, this layer verifies the user and then passes the request to the Presentation Layer. After receiving the output of the request it forwards the same to the client.

Apart from the Proxy Layer, all the other layers are hidden by firewalls. All the client requests and responses are captured by the proxy layer to ensure that the internal layers are not directly accessible from a public network.

### 3.2.2 *iMediK and Web-Services*

Web Services<sup>5</sup> are components accessible by Internet users through common Internet technologies and standards such as XML<sup>6</sup>, SOAP<sup>7</sup>, WSDL<sup>8</sup> and HTTP. The services offered by a Web Server should be first registered (such as in Universal Description Discovery and Integration [UDDI] registry) so that they can be exposed to the users. The iMediK server handles critical data of patients with different diseases and problems. In the iMediK environment, Web Services are offered through Webproxy Layer which in turn uses HTTPS to communicate with other layers to ensure adequate security. The interlayer communication is handled following SOAP standard. Being an integral part of iMediK, OGC at present is implemented as a component of the presentation layer and is not directly exposed as Web Services but it can be invoked through the Webproxy layer.

### 3.2.3 *Integration of OGC with iMediK*

OGC is a Java Applet, embedded in a web page. In the iMediK architecture (Figure 1), it resides as a tool in the Presentation Layer. When requested, it is loaded on the client window and executed by the browser's Java Virtual Machine (JVM). Once loaded, it communicates with the iMediK server using the HTTP protocol. It establishes an HTTP connection to the iMediK [6,7] server and then posts HTTP requests to it when any event such as annotating or drawing a circle is activated by a user. The HTTP requests are received and forwarded by the proxy layer to the presentation layer. At the presentation layer, the request is processed and then an HTTP request is made to the application layer which fetches the required information from the database layers.

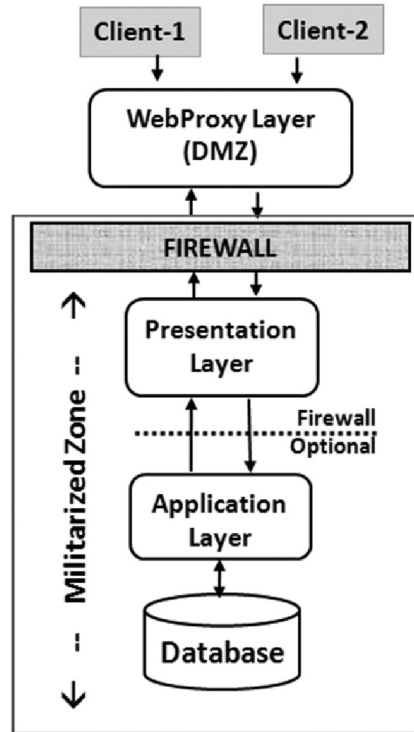
---

<sup>5</sup><http://www.w3.org/TR/ws-arch/>

<sup>6</sup>XML stands for Extensible Markup Language, <http://www.ietf.org/rfc/rfc3076.txt>, <http://www.w3.org/TR/xml/>

<sup>7</sup>SOAP is a simple XML-based protocol to let applications exchange information over HTTP, <http://tools.ietf.org/html/rfc3288>, <http://www.w3.org/TR/soap/>

<sup>8</sup>Web Services Description Language) is an XML-based language for describing Web services and how to access them., <http://www.w3.org/TR/wsdl>



**Figure 1.** Architecture of iMediK [6, 7]

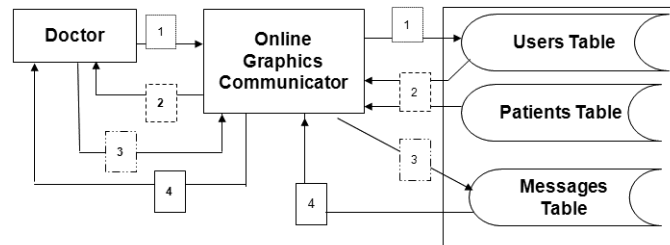
### 3.2.4 Components of OGC

The iMediK [6,7] server hosts database and ASPX<sup>9</sup> pages for interaction with the database tables. The overall message flow between the clients and OGC is shown in Figure 2. The following three tables are installed for storing data in the system:

- **User Table:** This table contains data of logged-in users. When a doctor logs-in, a message is posted to the server, and the table is updated. In the Figure 2, this message flow is represented by point 1.
- **Patient Table:** It contains a list of patients assigned to individual doctors. This table is not updated by the communicator who can only fetch data from it. This is populated by the iMediK server. In the Figure 2, this message flow is represented by point 2.
- **Message Table:** The message table is populated whenever an event is generated at the client end. In the Figure 2, this message flow is represented by point 3.

<sup>9</sup> ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It is the successor to Microsoft's Active Server Pages (ASP) technology. ASPX Pages are built in the ASP.NET environment. These pages are served by Microsoft's IIS. Refer: MacDonald, Matthew; Mario Szpuszta (2005). Pro ASP.NET 2.0 in C# 2005 (1st edition). Apress. ISBN 1-59059-496-7

The client also checks for any new messages that are intended for it. This flow is represented by point 4 in Figure 2.



**Message Flow between Users & Online Graphics Communicator.**

1. Login, Authenticate & Append
2. Get assigned patients. Select Patient, Get other doctors Assigned.
3. Start Send Messages and update MSG table
4. Check for any new messages.

**Figure 2.** Message flow between client and server

OGC comprises the following two major blocks:

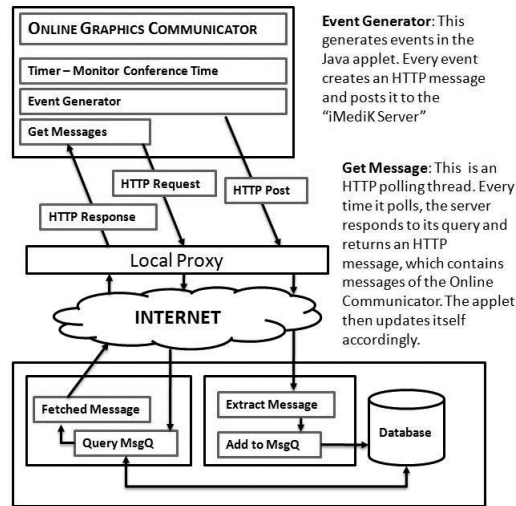
- **GetMessage Thread:** GetMessage thread polls (request for new data from) the server at regular intervals to check for any new messages and on receipt of a new message, the thread updates its state as per the message.
- **Event Generator:** When there is an event created by the connected user, a message for other clients is created and posted immediately. For example, if a user draws a line on the canvas, the message to draw the line is generated and subsequently the HTTP message is created and posted to the iMediK server.

OGC has been developed in Java<sup>10</sup> environment without using any extra/third party APIs<sup>11</sup>. It is an event driven system. For every specific event, a suitable HTTP message is generated. It is then posted to the iMediK server where the message is extracted and queued up for further processing. This system, however, does not function on push technology<sup>12</sup>. Hence the client polls the server at regular intervals to see if it has any message to process. Once a message is received, it is decoded and the system is updated accordingly. Since the system works based on polling strategy, updating would occur periodically based on the polling frequency.

<sup>10</sup><http://www.java.com/en/download/index.jsp>

<sup>11</sup>Third party (Application Programming Interface) API is a set of tools written and provided by other vendors (as freeware or shareware) which can be integrated into another system. A system that uses the APIs becomes dependant on those tools and the existing functions often become limited.

<sup>12</sup>Push technology, or server push, describes a style of Internet-based communication where the request for a given transaction is initiated by the publisher or central server. It is contrasted with pull technology, where the request for the transmission of information is initiated by the receiver or client.



**Figure 3.** Architecture of Online Graphics Communicator

Consultation between doctors over medical images is performed in a closed group fashion. For example, it is possible to have different groups of doctors using the same server concurrently discussing on different patients.

In this environment, if a client wants to open an image, it first selects and opens the image in its console. Then a message such as "01#CanvasNo#ImageFileID" is generated and sent to the server, which is then added to the Message Queue. Once the message is posted, the other connected clients can read their respective message queues. For example, when the message mentioned above is received by the client an image specified by "ImageFileID" in the canvas identified by "CanvasNo" is opened. Similarly the message, "2#canvasno#Color#start x#start y#end\_x#end\_y" is generated for drawing Straight Line, while the message "21#canvasno#percentage#startX#startY#width#height" is used for zooming a portion of canvas. Some of the sample text messages sent from a client to the server are given in Table 1.

**Table 1.** List of messages used in OGC communication

Action	Command
Straight Line	02#canvasno#Color#start x#start y#end x#end y
Ellipse/Circle	03#canvasno#Color#start x#start y#width#height
Rectangle	04#canvasno#Color#start x#start y#width#height
Freehand	06#canvasno#Color#x1#y1#x2#y2#x3#y3....
AnnotationText	07#canvasno#Color#text#Xposition#Yposition
Zoom Canvas	20#canvasno#percentage
Zoom Portion	21#canvasno#percentage#startX#startY#width#height
Text Message	13#text-message

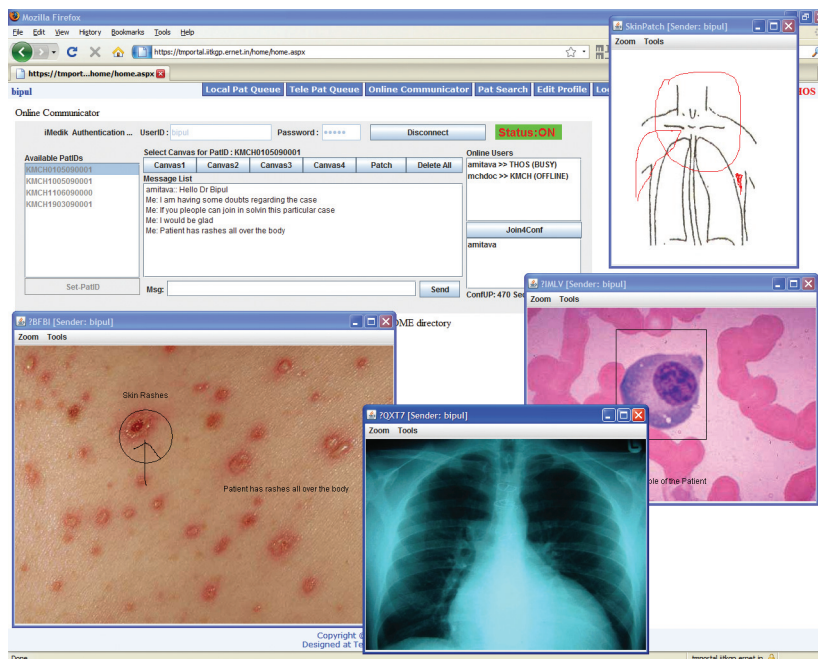


Actual processing of images such as drawing or annotating on the fetched image are performed by OGC on the client side. The server only stores and sends the data (images and messages) during each HTTP poll. The client on receiving the message does the parsing and changes its state accordingly. For example, if there is a message to close the canvas, the communicator closes the specified canvas immediately. The system works under the SSL<sup>13</sup> and hence the data transfer is secured.

#### 4. SAMPLE SYSTEM RUN

When a doctor logs in to the system, his/her information is posted. Accordingly, the User Table is updated and the user receives a list of patients assigned to him/her (fetched from the Patient Table). When a doctor selects a patient, the list of doctors whom he/she may consult for this patient along with their status is also provided. The assignment of doctors who can be consulted for a patient is assumed to have been decided beforehand and maintained in an appropriate table. The statuses of the doctors who can be consulted are categorized as:

- 1) **Offline:** User is not logged into the system.
- 2) **Online:** User is logged in, but not in conference.
- 3) **Busy:** User is in conference.



**Figure 4.** An example of OGC teleconsultation session

<sup>13</sup>Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide security for communications over networks such as the Internet. TLS and SSL encrypt the segments of network connections at the Transport Layer end-to-end. <http://www.ietf.org/rfc/rfc2246.txt>

The doctor initiating consultation selects a group of doctors from the permitted list who are currently online and invites them to join the conference for the specific case. Those who accept such a request become a party to the intending conference. Now all of these participating doctors can discuss among themselves the selected patient's medical images and records.

The client software posts new HTTP messages to the server whenever an event is generated. When a client generates any event such as drawing, text chatting, etc., an appropriate message is created and posted. This information is stored in the Message Table.

The client tools poll the servers to fetch new messages, if any, that may have been posted by other clients in the conference. At the end of a session, a user has to disconnect from the conference although he/she still stays connected to the server and can select a new patient or accept requests from other users. Message flow from the client to the iMediK server is shown in Figure 3. Currently, OGC is being used in a test bench environment with iMediK server at IIT Kharagpur as well as in another installation at Calcutta Medical College and Hospital, Kolkata, India. A typical OGC session is depicted in Figure 4.

In figure 4, the largest window shows the main interface for the users. It has different panels displaying the log-in process, available list of patients, message box, available online doctors with their statuses, etc. The system supports multiple display of images, graphics, etc. Windows may be created by a user when the specific command is issued.

Doctors intending to have consulting sessions with others first login into OGC. Once they login, on the right column they see a list of patients that are referred to by others. A doctor selects a patient whose images are to be discussed, and then on the left side, he/she gets a list of other doctors along with their statuses, who are assigned to the selected patient. Then the doctor sends an invitation to the online doctors. Doctors busy with other sessions will not receive the invitation. If the invitation is accepted, the user becomes a member of an OGC session. Now the participating doctors can perform online text and image chatting while a session is on. To start an online image session, a participating doctor selects an image and opens it on his/her OGC client. The other OGC clients receive this message almost instantly and they open the same image automatically. The image is centrally located in the Database Layer of the iMediK. The message contains the image details and the link to the image to be opened. The connected clients decode this link, and then send HTTP request to the iMediK server for the image to be opened. On receiving HTTP response, the image is opened on the fly without saving the image permanently on the disk. Now any doctor can mark a specific region of interest on the opened image by using any of the input tools provided. This time only coordinates and the command are posted to the clients. For example, if a line is drawn on the selected image, a command to draw a line is created and it is then posted to the server. During HTTP polling, on receiving such a command with appropriate coordinates, the receiving clients should draw the specified line on the opened image. Since the image is already opened, the marking is performed instantly on the opened image of the other clients. While one image session is in progress, the doctor can open other images of the patient as well.

The proposed OGC client can run on both Windows and Linux platforms. The

system has been thoroughly tested using the Firefox, IE, Opera and found to be working satisfactorily. The only requirement is that the browsers must allow Java Applets to send and receive data through HTTP requests. At the server end, we have used IIS<sup>14</sup> where .aspx<sup>15</sup> pages in webserver uses C# code to fetch relevant data from a patient database.

## 5. PERFORMANCES

Table 2 provides a very brief idea about the number of images each patient can have in a typical implementation of iMediK server in a hospital. The number of images and their content vary from patient to patient.

**Table 2. Images of real patients at a hospital**

Patient	Disease	Images
1	Urosurgery	9
2	Ophthalmology	6
3	Oncology	9
4	Dermatology	8
5	Tuberculosis	5
6	Oncology	9

The system tuning and performance testing was carried out with the iMediK server located at IIT Kharagpur and clients at different locations having different types of links as given below:

- **Net1:** This WAN<sup>16</sup> node is located at Kharagpur, India, but outside IIT Kharagpur's Network and is connected to 144kbps wireless USB modem provided by a service provider.
- **Net2:** This WAN node is located at Kharagpur, India, but outside IIT Kharagpur's Network and is connected to 2.4Mbps wireless USB modem provided by another service provider.
- **Net3:** This LAN<sup>17</sup> node is connected to the server by 100MBPS shared link within IIT Kharagpur LAN.

In the test bench a large number of messages of different sizes (depending upon the patient's records and images) were generated on different types of images as shown in Table 3, and the response time was measured.

<sup>14</sup>Internet Information Services (IIS) – formerly called Internet Information Server – is a set of Internet-based services for servers created by Microsoft for use with Microsoft Windows Operating Systems.

<sup>15</sup>ASPX Pages are built in the ASP.NET environment. Please refer to foot note number 9.

<sup>16</sup>A wide area network (WAN) is network that interconnects geographically-distributed computers or LANs.

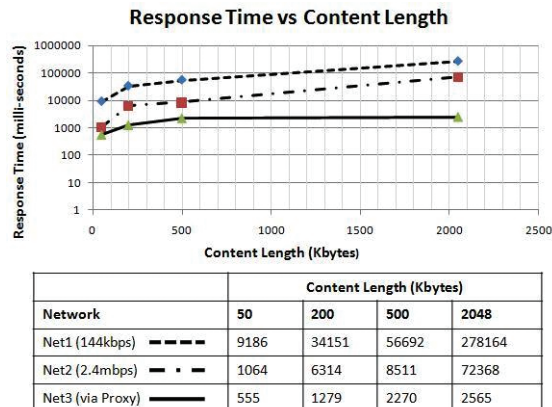
<sup>17</sup>A local area network (LAN) is a computer network covering a small physical area, like a home, office, or small group of buildings, such as a school, or an airport.

**Table 3. Image details used in OGC test bench**

Image	Resolution (pixels)	Size
1	2048_1536	393KB
2	1183_890	125KB
3	3123_2079	2.24MB
4	1120_1125	212KB
5	3109_799	629KB
6	513_599	41.5KB
7	900_900	275KB
8	1722_1430	167KB
9	1408_1240	130KB

The response pattern, in semi-logarithmic scale, is depicted in Figures 5. In Figure 5, message content mainly consisted of images with different sizes varying between 45K and 2.5MB. After an image is loaded, the response time reduces because only text messages are sent. The average content length of such text messages is usually less than 2048 bytes. In Figure 6, the initial image loading time for three images over different networks are exhibited. This figure also depicts the variation of message response time ( $T_x$ ) as a fraction of image loading time ( $T_y$ ) for different networks.

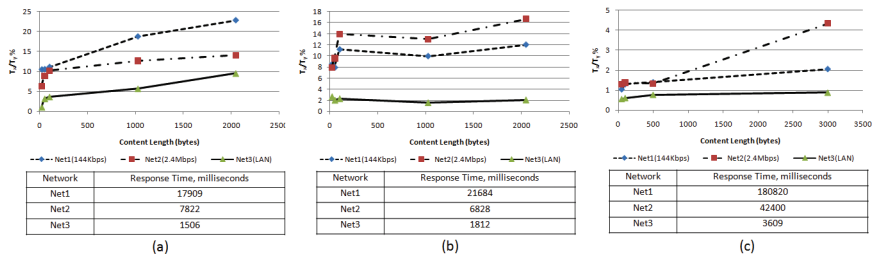
It may be observed in Figure 5 that response time increases with the image size. With high bandwidth channels (2.4 Mbps), the rate of increase in response time is not that high. For low bandwidth channels (144 kbps), the response time is quite high and significantly contributes to the initial time delay (shown in Figure 6) during online communication. Such long time delay may not be acceptable during live communication. Hence for very large images, it would be desirable to use store-and-forward technique, where the image is fetched in an offline mode before any online communication is initiated.



**Figure 5.** Response time (in milliseconds) of various image sizes (in Kbytes) in different networks.

The system has also been tested with clients distributed across India at different locations. It was observed that as long as the image/message size is not very large, the quality of online consultation even with 144kbps is acceptable.

It was also observed that even with limited bandwidth provided by data-cards and USB dongles (e.g. around 144 kbps), the communication delay with text messages over images already fetched was within acceptable limits (Figure 6). The main advantage of this system is in providing a platform for independent image consultation over the Internet. Remote consultations can be readily achieved from virtually anywhere over different types of communication links. The bandwidth limitation, however, is a major constraint. As expected, the response time is faster with higher bandwidth networks than with the lower ones. The system can also be used to make certain measurements on a displayed image (such as with ECG signals) or on a portion of it, and send the results to the doctors.



**Figure 6.** Response-to-load-time ratio ( $T_x/T_y$ ) vs. content length for different networks.

(a) Image size 128 Kbytes. (b) Image size 217 Kbytes. (c) Image size 2.36 Mbytes.

## 6. CONCLUSION

This paper describes a simple yet powerful tool OGC to support online teleconsultation among doctors who participate in discussing a patient's medical images. It addresses the issues related to transfer of images and messages, and reduction of data redundancy during data transfer. It relies only on HTTP/HTTPS protocol which is always permitted in any webserver environment. A version of the OGC has been integrated with the iMediK server and is hosted on <http://tmportal.iitkgp.ernet.in/>. This system currently deals with still images and image profiles only. Teleconsultation with stored videos or live videos is yet to be addressed.

## ACKNOWLEDGMENT

This work was carried out under a sponsored project supported by the Ministry of Communication and Information Technology, Govt. of India (refer Grant no. 1(23)/2006-ME&TMD, Dt. 07/03/2007).

**REFERENCES**

- [1] W. Cai, D. D. Feng, and R. Fulton, "Web-based digital medical images," *IEEE Computer Graphics and Applications*, vol. 21, no. 1, pp. 44–47, 2001.
- [2] J.-S. Lee, C.-T. Tsai, C.-H. Pen, and H.-C. Lu, "A real time collaboration system for teleradiology consultation," *International Journal of Medical Informatics*, vol. 72, pp. 73–79(7), December 2003.
- [3] L. Ling, Y. Dezhong, L. Jianqing, L. Bin, and W. Ling, "A multimedia telemedicine system," in *Proceedings of Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, China, 2005, pp. 3746–3748.
- [4] S. K. Yoo, K. M. Kim, S. M. Jung, K. J. Lee, and N. H. Kim, "Design of multimedia telemedicine system for inter-hospital consultation," in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, San Francisco, CA, USA, 2004, pp. 3109–3111.
- [5] H. Huang, Ed., *PACS: Basic Principles and Applications*. New York: Wiley-Liss Publisher, 1999.
- [6] A. K. Maji, A. Mukhoty, A. K. Majumdar, J. Mukhopadhyay, S. Sural, S. Paul, and B. Majumdar, "Security analysis and implementation of web-based tele-medicine services with a four-tier architecture," in *International Workshop on Connectivity, Mobility and Patients' Comfort (CMPC)*. Tampere, Finland: IEEE Computer Society Press, 2008, pp. 46–54.
- [7] A.K. Maji, A. Mukhoty, A. K. Majumdar, J. Mukhopadhyay, and S. Sural, "Secure healthcare delivery over the web: A multi-tier approach in india," in *Proceedings of the Indian Conference on Medical Informatics and Telemedicine (ICMIT 2006)*, India, 2006, pp. 62–67.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

