

The Reliability of the XMPP Protocol Extensions as a File Transfer Mechanism in Dedicated Healthcare Networks

Kristian Andreassen, Johan Gustav Bellika

University of Tromsø, Department of Computer Science, Medical Informatics and Telemedicine group, Norway.

ABSTRACT

Objective: File transfer is a basic service that needs to be provided by a network dedicated to telemedicine and ehealth use. Store and forward solutions have so far been based on protocols such as SMTP/POP (Simple Mail Transfer Protocol/ Post Office Protocol), FTP (File Transfer Protocol) and HTTP (Hypertext Transfer Protocol). However, these are not well suited to the transfer of large files that are associated with healthcare data. To meet the growth in number of institutions connected to the Norwegian Health Network and the amount of network traffic between health institutions within the network, there is a need for better, more reliable and scalable solutions for file transfer. For example the current requirement to support telemedicine applications is transfer of files within the range of 1 to 200 MB for series of CT (Computerised Tomography) or MRI (Magnetic Resonance Imaging) scans.

The objective of this study was to investigate the reliability and the potential of the XMPP (Extensible Messaging and Presence Protocol) file transfer extensions as the underlying file transfer mechanism for telemedicine solutions in a dedicated healthcare network.

Methods: We deployed clients and servers for testing purposes in a real health network setting. We tested the XMPP protocol extensions in three different configurations by transferring files with sizes ranging from 1 to 200 MB. We also tested the FTP and SMTP/POP protocols to have a realistic basis for comparison.

Results: The results of our test shows that in the production environment the XMPP protocol extensions configurations have a possible transfer rate ranging from 0.95 MBps to 0.037 MBps depending on the configuration and protocol extensions used. In comparison the FTP transfer rate was 0.8 MBps and for SMTP/POP was 0.017 MBps. The configuration with the lowest transfer rate also has the highest potential for transfer error. A favourable feature of XMPP is that files are not stored on a third party server as with current file transfer protocols for FTP and SMTP/POP, but only on local area networks. This is perceived as more secure and reduces the costs associated with security maintenance.

Conclusion: The current file transfer services for store and forward telemedicine is based on use of the SMTP/POP protocols which is not suitable for file attachments of the size used in this study. The other alternative, the FTP protocol, has good transfer efficiency but has other drawbacks such as maintainability and the need for server and client configuration. Based on the results of our study we conclude that the reliability and efficiency, and the need for configuration and maintenance of the XMPP protocol extensions, indicates that a file transfer service based on these extensions is feasible.

Correspondence and reprint requests: Johan Gustav Bellika, University of Tromsø, Department of Computer Science, Medical Informatics and Telemedicine group, Norway. E-mail: gustav@cs.uit.no.

INTRODUCTION

The Norwegian Health-Network (NHN) is the main supplier of infrastructure and communication services between primary and specialist healthcare providers in Norway. The most important form of electronic collaboration is communication of medical information to and from hospitals. Examples of data communicated include referral letters, laboratory results, radiology results and discharge letters. This communication is performed today mostly by using the well-known email protocols POP (Post Office Protocol) and SMTP (Simple Mail Transfer Protocol), but these have limitations for sending large files as attachments¹. This is becoming increasingly necessary as solutions in healthcare evolve with the production of high resolution digital images and advanced electronic medical documents. Transmitting these documents and data files between communicating parties will play a vital part in modern healthcare. Although problems with transmission could be addressed by increasing the capacity of communication channels, the development of new technologies and new protocols offers other opportunities.

The Extensible Messaging and Presence Protocol (XMPP) is an open XML (extensible markup language) technology for real-time communication². It includes a wide range of applications such as instant messaging, presence, media negotiation and generalised XML routing. Some of these features might be beneficial in the development of efficient and secure communication within a health network. The objective of this study is to identify if one or more of the XMPP protocol extensions could be used as file transfer tools and deliver better telemedicine services.

METHODS

The XMPP technology is based on streaming XML³ and a large number of extensions (XEP's) to the base protocol have been developed⁴. Some of these extensions have been aimed at extending the XMPP technology to support file transfer. Two of these extensions – XEP 0065 Socks 5 Bytestreams⁵ and XEP 0047 In-band bytestream⁶ were tested during this study.

Using these two extensions it is possible to create three configurations for file transfer as shown in Figure 1:

- (i) Peer to peer
- (ii) Proxy transfer
- (iii) In band

In these configurations the clients are located inside the health institutions, typically behind firewalls, and the XMPP server is located in the health network. Figure 1a shows the use of XEP 0065 where the client itself acts as streamhost for the file transfer (from right to left). In Figure 1b, also based on XEP 0065, the proxy acts as streamhost for a mediated file transfer (from left to right). In Figure 1c the file

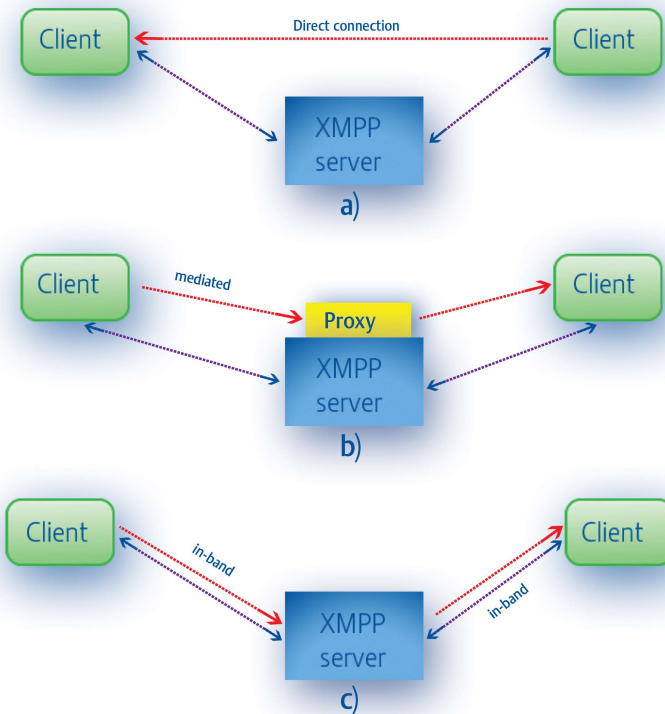


Figure 1. Three configurations for file transfer using XMPP protocol extensions:

- a) XEP 0065 direct (peer-to-peer) connection using the initiating XMPP client as streamhost,
- b) XEP 0065 mediated transfer using a proxy as streamhost, and
- c) XEP 0047 In-band bytestream transferring data via the XMPP server

transfer is done “in-band” by adding base64 encoded data into the existing XML stream between the clients through the XMPP server according to the XEP 0047 protocol extension.

It should be appreciated that these three connections are being tested in this study to evaluate the speed of file transfer for each of them individually. In practice, some of these connections may not be possible due to local and national restrictions placed on how connections into a health institution’s computer network can be established. In Norway for example the configuration shown in Figure 1a is not legal because this configuration makes it possible for open network connections from a less secure zone into the secure zone within the health institution’s local area network. Common to all the configurations shown in Figure 1 are that files are not stored on a server within the health network, which is perceived as a less secure zone compared to local area networks within individual health institutions. The

Table 1. Hardware used during the tests

Environment	Component/Name	Specification	Operating System (OS)
Pilot	Computer A	Intel Core 2 CPU 4400 @ 2.00GHz, 3.5 GB RAM	Windows XP
Pilot	Computer B	Intel Core 2 CPU 3300 @ 1.74GHz, 1.0 GB RAM	Windows XP
Pilot	Computer C	AMD Duron II @ 750 MHz, 512 MB RAM	Windows XP
Pilot	Computer D	AMD Athlon 2200 @ 1.4GHz, 396 MB RAM	Windows XP
Pilot	Server	Intel Core 2 CPU 4400 @ 2.00GHz, 3.5 GB RAM	Ubuntu 7.04
Production	Computer A/Radiology A	Intel Core 2 CPU 4400 @ 2.00GHz, 3.5 GB RAM	Windows XP SP2
Production	Computer A/Radiology A	Intel Core 2 CPU 4400 @ 2.00GHz, 3.5 GB RAM	Windows XP SP2
Production	Computer A/Radiology A	Compaq P4-630 @ 1.0GHz, 512 MB RAM	Windows XP SP2
Production	Computer A/Radiology A	AMD Athlon 2200 @ 1.4GHz, 396 MB RAM	Windows XP SP2
Production	Server	Compaq P4-630 @ 2.0GHz, 512 MB RAM	Windows XP SP2

CPU = Central Processing Unit, RAM = Random Access Memory, SP = Service Pack.

current protocols used for file transfer, SMTP/POP and FTP need to store messages on a sever until it is collected by clients polling the server from within their health institution. However, if clients stop polling the server for available messages/files, the messages/files will not be removed from the servers.

To perform the tests we used the hardware shown in Table 1. The network capacity on a wired connection setup has a theoretical capacity of 100MBps, based on the switch and cable components. However, during the development of a test environment, it soon became clear that many of the available software solutions for both server and clients did not support the protocols and extensions needed. This is mainly due to the fact that the available clients were tuned towards a user base which is intended for chatting. On the server side, there were similar issues when trying to find the best real time collaboration (RTC) software for the study. Once again most of the available implementations were focused on delivering the best possible experience for instant messaging. For the study the software chosen was Openfire² for the server installation and Spark⁸ for the chat client. Both software solutions are developed by the Jive Software organisation, and are released for free. Most of the RTC implementations available do support the extensions needed for this study, so

the choice of software was based on personal communication with developers and experienced users of the XMPP technology.

To perform the measurements we deployed a test setup using the actual operating environment running within the NHN. At the pilot installation we tested the configurations in an isolated environment to make sure that all the components used in the tests worked as intended. The setup scenario for the study consisted of two radiology departments and a communication link, with the XMPP server being used to transfer files from one department to the other. Using FTP in its basic format required a similar setup with FTP clients and a FTP-server to relay the files. For the FPT test we used NcFTPd for the server and SmartFTP for the client software. The SMTP/POP test was performed by installing Postfix with accompanying tools on the server and sending a series of messages with file attachments. Tests were performed by repeatedly (fifteen times) transferring a 102400KB file for each of the 3 XMPP configuration protocols and also for comparison using FTP and SMTP/POP protocols. The average time/speed to transfer the files together with any errors were measured. Testing was subsequently performed in the production environment with a variety of file sizes to measure the success/failure ratio, the scalability of the configurations, resilience to errors and efficiency.

RESULTS

The file transfer results from the pilot test are show in Table 2. The results were affected by the limited bandwidth available between the server and the clients which are favourable for FTP and the direct (peer-to-peer) high speed connections used in the pilot

Table 2. *File transfer performance – pilot environment*

Setup	File size (KB)	Number of errors	Number of transfers	Average time	Average speed	From/To Computer
Direct (peer- to-peer)	102400	0	15	11.65 s	0.8 MBps	A-B & A-C & A-D B-C & B-D & D-A
FTP	102400	0	15	17.2 s	0.8 MBps	A-Server & B-Server
Mediated proxy	102400	0	15	21.2 s	0.3 MBps	A-B & A-C & A-D B-C & B-D & D-A
In-band	102400	0	15	5 min 44 s	0.02 MBps	A-B & A-C & A-D B-C & B-D & D-A
SMTP/POP	102400	3	5	9 min 37 s	0.017 MBps	D-Server-A

environment and are unfair towards the remaining transfer methods. The remaining results may also be affected by temporary bandwidth fluctuations in the pilot environment. The 3 errors experienced in the SMTP/POP pilot test were as a result of time-out for the outgoing message. For the pilot test performed on the in-band configuration, the average transfer time taken was 5 minutes and 44 seconds, giving an average transfer speed of 0.02MBps for transferring a 102400KB test file. This indicates that in-band configuration imposed a high load on the server.

Table 3 shows the results in the production environment for the direct (peer-to-peer) and mediated connection options, and Table 4 for the In-band bytestream testing. The peer to peer file transfer was found to have the best overall performance with respect to speed and time consumption. For the in-band bytestream when transferring a large file (>100 MB) using a Java Virtual Machine (JVM) heap size of 64MB, 4 of the transfers failed and all of the transfers of a 200MB test file failed. However, after adjusting the JVM heap size to 512MB all 10 transfers of the 200MB test file succeeded. Files still however took a long time to transfer

Table 3. File transfer performance for XMPP peer to peer and proxy mediated transfer in the production environment

Setup	File size (KB)	Transfers	Failed	Av Speed (Mbps)	Av time (s)	CPU / JVM memory load	From/To Computer
Direct	51200	25	0	0.95	52.3	4% / 11%	A-B & B-A
Direct	102400	25	0	0.87	114	3% / 10%	A-B & B-A
Direct	204800	25	0	0.89	223	4% / 12%	A-B & B-A
Mediated	51200	25	0	0.84	59.3	6% / 17%	A-B & B-A
Mediated	102400	25	0	0.78	127	8% / 19%	A-B & B-A
Mediated	204800	25	0	0.82	243	6% / 18%	A-B & B-A

Table 4. File transfer performance for XMPP in band bytestream transfer in the production environment

File size (KB)	Actual size transferred (KB)	Transfers	Failed	Av Speed (Mbps)	Av time (s)	Java Heap Space	CPU/ JVM memory load	From/To Computer
102400	138235	10	4	0.035	3863	64	11%/80%	A-B & B-A
204800	278528	10	10	0	0	64	14%/100%	A-B & B-A
51200	69240	10	0	0.053	1289	512	6%/9%	A-B & B-A
102400	138235	10	0	0.037	3652	512	6%/9%	A-B & B-A
204800	278528	10	0	0.038	7480	512	6%/11%	A-B & B-A

(approximately thirty times as long as peer to peer or mediated transfer for a 100 or 200 MB file).

DISCUSSION

We have demonstrated the feasibility of using XMPP protocol extensions to transfer large files, and obtained favourable results with respect to transmission times for the peer to peer protocol. However inevitably for a system to be implemented into practice a number of other factors other than just speed of transfer need to be taken into consideration. Some of these include:

- Control of secure traffic
- Scalability
- Security of transferred files

Control of Secure Traffic

A health network consists of several firewalls controlling access to and from the computers and systems. Any new organisation (service) added to the network must be able to pass traffic to and receive traffic from other organisations (services) in the network. This is achieved through dedicated ports connecting the organisation to the network. A new service is classified by its need for open ports into the categories of zero, one or several. This classification affects both the actual work that needs to be done and the security and risk analysis that need to be done prior to introducing a new service. If the in-band alternative is used it would need only 1 port for transferring files because files are transferred in-band. The other option for file transfer, SOCKS5 would need to use the proxy option for the file transfers because it would not be possible to use the direct-connection if either of the clients was placed behind a firewall. For the proxy alternative the communication and control stream is still performed over the standard ports defined by XMPP. This is in many ways similar to the FTP alternative which uses one port for its control stream, with the actual data transfer done over another port depending on the file transfer mode. This isn't very different from the SOCKS5 proxy solution implemented in XMPP and delivers the same amount of work related to opening ports in firewalls. The two extensions used for file transfer in this study have also been bundled into an extension called XEP-0096 File Transfer. This extension implements seamless file transfer giving the client a fallback option if the preferred choice of transfer method doesn't comply. This does in fact start the file transfer using the inband solution if any of the two SOCKS5 methods fails. Even though the in-band solution would perform slowly, the file would reach its intended recipient.

Scalability

Adding more departments as senders and receivers start using XMPP would require adding a new address (known as a Jabber ID or JID) to the roster of each XMPP

client. This roster can be stored and kept updated by the presence functionality built into XMPP. By adding a new department, the transmitting XMPP client has to address the file with another JID, before file transfer is possible. Using FTP this would be a bit more complicated. Since a file transmitted from Radiology A to Radiology B would have to be addressed in some way, one would probably have to use more than 1 FTP-server to achieve this. Each FTP-client has their own user account on the FTP-server, allowing them to login and upload or download a file. The client itself doesn't by default know who will be downloading the file after it is stored on the FTP-server. This means that the downloading client will have to be tuned towards looking for a unique identifier in the filename, or to check a unique account on the FTP-server, pertaining to each FTP-client. This would result in a complex structure if an N:N relationship is necessary. It would probably be possible to develop dedicated and automated solutions for transferring files without having these scalability issues related to the base functionality of FTP. XMPP also has its scalability issues, especially related to presence overhead. A lot of data is transmitted through the XMPP server to keep the statuses of all clients updated on all connected clients. In the study setup the presence overhead was unnoticeable because the number of clients was very small. When the number of connected clients reaches 500–1000 the amount of presence data communicated through the XMPP server would be noticeable.

Security of Transferred Files

A major advantage associated with the use of XMPP as the transfer solution, is its lack of centralised file storage. If departments base their dedicated file transfer mechanism on XMPP, any files transferred would never be stored on any third part location. They would be transferred directly from the transmitting department to the receiver's department. If the receiver isn't available the transmitter will be notified of this by the built in presence functionality.

Using FTP in its basic form, all files need to be stored on a FTP-server until the receiver downloads them. After the receiver has completed the download, the file needs to be deleted in a safe and non-reconstructable manner. If the receiver for some reason should experience any trouble, the file will be stored on the FTP-server. The risk of data loss or unauthorised access might not be any bigger on the FTP server than other storages. However, an intrusion on a centralised storage unit would make a lot more data accessible to the intruders. Centralised storage of files consequently has to be placed under very rigid and strict security measures which increases the cost related to running the service.

CONCLUSION

Given the need to transfer large files, the in-band solution does not achieve the needed levels of performance. It performs too slowly and the load on both the hard-

ware and software used is unacceptably high. However for smaller files and even non-urgent transfers, the in-band solution has some elements that give it a possible level of acceptance. As a solution for transferring large files in any network setting, it does not meet the needed level of performance. The mediated proxy (SOCKS5) solution has its limitations when it comes to network security like firewalls and network enabled (NAT-ed) devices. It relies on a proxy placed outside the LAN's involved in the communication link, or that the clients transferring the files have a possibility of publishing its network information. In a NAT-ed network this will not be possible, and a proxy is the only solution. Its performance is satisfactory compared to both the pre-set requirements and other solutions for file transfer. XMPP therefore delivers several positive and new aspects that makes it feasible as an alternative for a file transfer solution in telemedicine and ehealth dedicated networks. The protocol also defines a new and interesting option as the aspect of communicating "presence" and there are no doubt other undiscovered possibilities regarding the use of XMPP.

REFERENCES

- 1 Ewasil W, Plank J, Beck M, Wolski R. *IBP-MAIL: Controlled Delivery of Large Mail Files*. Paper presented at the NetStore '99: Network Storage Symposium, 1999.
- 2 Jabber.org Team. (2005). Jabber – Overview. <http://www.jabber.org/about/overview.shtml>.
- 3 Saint-Andre, P. Streaming XML with Jabber/XMPP. *Internet Computing, IEEE*, 2005; **9**: 82–89.
- 4 XMPP Standards Foundation. (1999–2008). XMPP Extensions. <http://www.xmpp.org/extensions>.
- 5 Smith, D., Miller, M., & Saint-Andre, P. (2007). XEP-0065: SOCKS5 Bytestreams [Electronic Version], 1.7. <http://www.xmpp.org/extensions/xep-0065.html>
- 6 Karneges, J. (2006). XEP-0047 : In-Band Bytestreams (IBB) [Electronic Version], 1.1. <http://www.xmpp.org/extensions/xep-0047.html>.
- 7 Jive Software – OpenFire.
- 8 Jive Software – Spark.

