Comparative Performance Investigation of DICOM C-STORE and DICOM HTTP-based Requests

Amandine Le Maitre, Jude Fernando and
Yannick Morvan
b-com Institute of Research and Technology,
Rennes, France
`amandine.lemaitre@b-com`

Gilles Mevel and Emmanuel Cordonnier
ETIAM, Rennes, France
`gilles.mevel@etiam.com`

Abstract— **Increasingly, physicians have to access clinical images distributed over multiple healthcare organizations. To this end, two DICOM protocols may be used: a regular DICOM C-STORE transaction or an HTTP-based DICOM request such as WADO or STOW. A major problem of the DICOM C-STORE transaction is that it is inefficient to transfer DICOM data sets that consist of thousands of DICOM objects (such as functional MRI data set) because of the large number of negotiations involved in the transfer. We compare the performances of C-STORE transactions with the STOW HTTP-based protocol, and show that the STOW protocol can divide the transfer time by about 50 when compared to a DICOM C-STORE transaction for studies that consists of thousands of DICOM objects.**

I. INTRODUCTION

The development of internet technologies is transforming the way hospitals access, share, visualize and archive clinical data. This development is especially reshaping the borders of hospital information systems and is transforming isolated departmental data repositories into interconnected hospital networks. More specifically, these isolated data repositories increasingly have to connect to each other and share patient records to separate hospital departments or even to distant clinical enterprises. For example, sharing patient records is especially important for collaborative clinical processes such as Multi-Disciplinary Team (MDT) meetings, during which multiple physicians from various clinical organizations discuss a diagnostic and treatment of a patient.

One can distinguish two approaches for sharing clinical data. A first method is to employ *centralized* cross-enterprise repositories. A major advantage of this approach lies in its simplicity[1]. However, these centralized repositories involve major investments in infrastructures. Additionally, the clinical organization loses control over patient data. Finally, sharing clinical data outside the border of the hospital mitigates the security and increases the complexity of the solution deployment to fulfill regulations such as Health Insurance Portability and Accountability Act (HIPAA) in the US. A second approach is to interconnect data repositories *distributed* over multiple enterprises. This second approach has multiple advantages. First, it allows clinical organizations to maintain control over patient data. Additionally, a distributed environment lowers entry barriers to network-based patient care. In this work, we focus on the second approach that is aiming at sharing images distributed across multiple clinical organizations. In that context, a framework that allows physicians to share images across multiple organizations is the Cross Enterprise Document Sharing for Imaging (XDS-I) IHE profile (Integrating the Healthcare Enterprise) [2]. In order to share and transfer DICOM data set, this XDS-I integration profile allows two possible protocols that are (1) a regular DICOM C-STORE or (2) an HTTP-based STOW request.

A major problem when using DICOM C-STORE is that it is very inefficient to transfer a large amount of DICOM objects. For example, let us consider a functional MRI data set that consists of $50,000$ slices/DICOM objects for a size of 400 MB. The transfer of such a data set using C-STORE messages may take several hours over a DSL connection. One reason explaining such a high delay is the necessary negotiation performed prior to transferring DICOM objects. In this paper, we investigate and evaluate the transfer of DICOM data set of various size using C-STORE and STOW. Experimental results show that an HTTP-based request can divide by about 50 the transfer time when compared to a regular DICOM C-STORE.

The remainder of this paper is organized as follows. Section II describes the general XDS/XDS-I framework for exchanging clinical document and images between clinical organizations. Section III provides details about the C-STORE and STOW protocols and discuss their respective advantages. Experimental results are provided in Section IV and the paper concludes in Section V.

II. CROSS-ENTERPRISES CLINICAL IMAGES SHARING

As a first step towards sharing clinical records between clinical organizations, the Cross Enterprise Document Sharing (XDS) IHE profile (Integrating the Healthcare Enterprise) was defined. Using the XDS profile, a document is shared by, first, uploading the file to a *repository* and, second, indexing and storing the metadata in a *registry*. Clinicians can subsequently query the registry to retrieve documents from the repository. However, this approach is inefficient to handle clinical images because of the impractical duplication of images in the repository [3]. To address that problem, the so-called Cross-enterprise Document Sharing for Imaging (XDS-I and XDS-I.b) was proposed [2]. The idea followed by XDS-I is to hold searchable metadata in the registry, itself linking to a so-called *manifest* stored in the repository. That

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

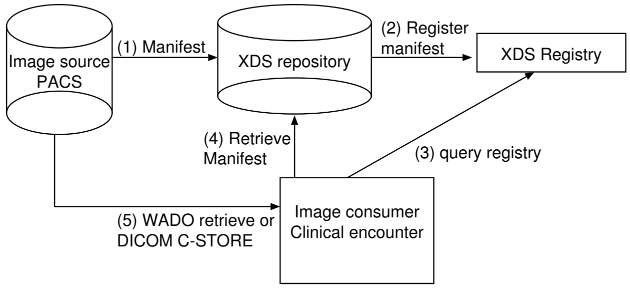

Fig. 1. Architecture overview of the Cross-Enterprise Document Sharing for Imaging (XDS-I)

manifest avoids the unnecessary duplication of images by providing the original location of images (see Figure 1). The manifest corresponds to a Key Object Selection [4] document providing the list of DICOM objects and their respective location. The XDS-I integration profile allows either a DICOM connection to a local Picture Archiving and Communications Systems (PACS) or an HTTP-based DICOM request (See Figure 1). We discuss both protocols in the sequel.

a) **DICOM connection:** According to the IHE XDS-I profile, image retrieval can be performed using regular DICOM transaction such as DICOM C-STORE. A major disadvantage of a DICOM connection is that it requires (a) a fixed and public IP address, (b) an Application Entity (AE) Title and (c) a port number [4]. This constitutes a significant issue because the IP of the servers are often dynamic and servers are not visible outside the hospital network. Note that a usual solution to address above mentioned problems is to deploy a specific Virtual Private Network (VPN) that provides a secure communication between hospitals. These VPN's are unfortunately difficult to set up, slow and do not constitute viable solutions.

b) **HTTP-based WADO or STOW requests:** Alternatively, the IHE XDS-I proposes to employ HTTP-based requests such as WADO. Such a solution is especially supported by the recent development of the Cross-Community Access for Imaging (XCA-I) profile [5]. This XCA-I profile aims at sharing and accessing imaging documents across multiple hospital/enterprises. This profile is especially focusing on sharing imaging documents using web service and addresses issues related to regular DICOM connections. As a first step towards a full implementation of the XCA-I profile that relies on a Web Services-based WADO, we focus and investigate performances of a HTTP-based DICOM RESTful implementation in a push scenario.

Note that the manifest stored in the XDS-I repository contains only the DICOM AE Title of the PACS but does not specify the IP address and port. As a result, DICOM transactions such as C-STORE cannot be easily performed across multiple hospitals/enterprises.

III. DICOM OBJECTS TRANSFER METHODS

In this section, we provide details about two DICOM protocols employed for transferring DICOM objects. These include (1) the regular DICOM transfer protocol (C-STORE and C-MOVE) and (2) the HTTP-based DICOM requests.

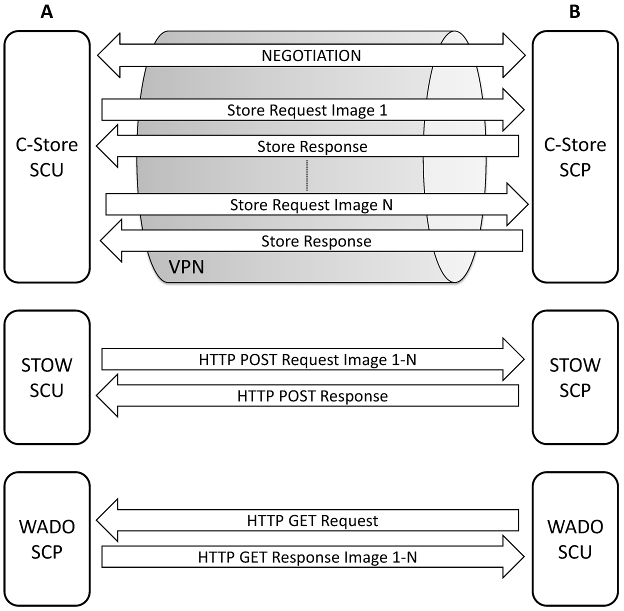

Fig. 2. General illustration of DICOM objects transfer scenarios from A to B.

A. Transfer using regular DICOM messages

We now present the DICOM C-STORE and C-MOVE. The C-STORE service is used to store a DICOM object on a so-called Application Entity (AE). This AE is employed to identify a DICOM node or server on a network. The Service Class User (SCU) initiating the transfer sends a C-STORE message followed by the object to be stored. Next, the Service Class Provider (SCP) stores the object and sends a C-STORE response. Only one object can be transferred per C-STORE message (see Figure 2). The C-MOVE message is used to pull or retrieve DICOM objects. More specifically, the SCU sends a request to the SCP to retrieve DICOM objects. Those objects are locally retrieved [1] by the SCP and subsequently stored by a C-STORE message on the SCU. Finally, a C-MOVE response is sent at the end of the transfer.

B. Transfer using HTTP-based DICOM requests

We now present the HTTP-based requests DICOM-STOW and DICOM-WADO. STore Over the Web (STOW) is a DICOM protocol that *sends or posts* DICOM objects using the HTTP protocol [6]. In order to send DICOM objects, HTTP POST requests are performed. The body of the request consists of a multipart/related content with one DICOM binary object per part [7]. Upon completion of the reception, an HTTP response to POST is sent back. This response consists of the HTTP status and contains an XML body listing received and missing objects and potential failure reasons (see Figure 2). Web Access to DICOM persistent Objects (WADO) is a DICOM protocol used to *get or*

---

[1] DICOM files may be locally stored on the SCP in an SQL database, and thus locally retrieved using SQL queries.

retrieve DICOM objects [8]. Similarly to DICOM STOW, in order to retrieve a DICOM object, an HTTP GET request is performed. The objects can be retrieved by study, series or instance. To define which objects should be retrieved, the Unique IDentifier (UID) of the object is employed and specified in the URL of the GET request. As a response, the WADO server sends an HTTP response with a multipart/related content type, where each part of the multipart is a DICOM binary object (see Figure 2).

IV. EXPERIMENTAL RESULTS

In this section, we describe the performed experiments and present the results. We especially focus on the C-STORE and STOW protocols. WADO transfer performances will be evaluated in a future work. However we expect similar results since the objects are transfered the same way in both protocols.

A. IMPLEMENTATION DETAILS

We implemented the STOW protocol using the POCO library [9]. POCO is an open-source C++ library dedicated to network programming. It features the advantages of being easy to deploy and being supported by multiple platforms. Moreover, our implementation is compact and has a limited number of software dependencies.

To perform a fair comparison, we slightly modified the implementation of C-STORE in DCMTK [10]. More specifically, C-STORE initiates the transfer by negotiating an association. For that negotiation, a presentation context is constructed by parsing the DICOM objects to determine the Service-Object Pair (SOP) classes and the transfer syntax that shall be employed. However, this parsing constitutes a time consuming operation and is not done when sending images using DICOM STOW. We therefore modified the C-STORE implementation of DCMTK to provide SOP classes and transfer syntax as an input. That implementation was used with appropriate parameters so that images are not parsed by the SCP.

B. EXPERIMENTAL SETUP

We performed two experiments to compare the transfer performances of the DICOM C-STORE with the DICOM STOW protocols. For this evaluation, we employed a network with a transfer speed of 100 $Mbit/s$. DICOM objects were transferred between two workstations using (1) a STOW request and (2) a DICOM C-STORE service. For each of these transfer methods, timers are inserted at the client side application. In the case of a STOW-based transfer, the timer is started at the beginning of the request and stopped when the response is received. In the case of a regular C-STORE-based transfer, the timer is started before the association negotiation and stopped when the response is received. Both experiments especially focus on evaluating the efficiency of the protocols when transferring large amounts of DICOM objects. As a first step, we evaluated the performances of both protocols using typical data sets. In a second step, we especially focus on the transfer of a functional MRI data set that consists of a large number of DICOM objects (55,890 objects).

Type of DICOM object	Size of DICOM objects	Number of DICOM object	Total size
Compressed CT-scan	98.6 kB	889	83.7 MB
Un-compressed CT-scan	514.7 kB	889	446.9 MB
DICOM-MPEG scan	1.1 GB	1	1.1 GB
MRI-scan	6.9 kB	55,890	377 MB

TABLE I
PROPERTIES OF EACH DICOM DATA SET

Experiment 1: transferring typical data sets: The first experiment evaluates the performance of each transfer protocol using multiple types of data sets. For each transfer type, the following data sets were used: (1) a compressed CT scan, (2) an uncompressed CT scan, (3) a DICOM data set embedding an MPEG video, and (4) a functional MRI scan. Note that the compressed and uncompressed volumes correspond to the same original volume. An important aspect of each of these data sets is that they consist of a varying number of files/objects and thus a varying number of acknowledgements. Table I summarizes the properties of each data set.

These data sets are of interest for the following reasons:

1) un-compressed CT scan: to evaluate the transfer rate of a typical DICOM data set,
2) compressed CT scan: to evaluate the influence DICOM object sizes.
3) DICOM MPEG: to evaluate the transfer rate of a small number of large DICOM objects,
4) functional MRI: to evaluate the transfer rate of a large number of small DICOM objects,

Experiment 2: transferring functional MRI data sets of varying size: The second experiment investigates the transfer of functional MRI data sets which consist of a varying number of DICOM objects. More specifically, increasing the number of DICOM objects affects the transfer efficiency because it involves a higher number of DICOM acknowledgement (see Figure 2). Therefore, we created 5 sub data sets using the original functional MRI data set, which consists of $55,890$ DICOM objects, by splitting in 5 equal parts. Each of these part/data sets consists of $11,178$, $22,356$, $33,534$, $44,712$ and $55,890$ objects, respectively. The transfer was performed using C-STORE DICOM messages and STOW request.

C. RESULTS AND DISCUSSION

We now discuss the results obtained in the two previously described experiments.

Experiment 1: transferring typical data sets: Figure 3 illustrates the transfer time of each data set using C-STORE and STOW. From the figure it can be noted that the transfer

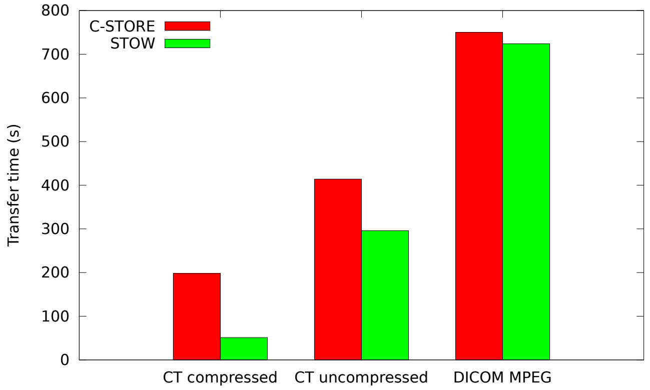

Fig. 3. Comparison of transfer speed for different types of DICOM objects

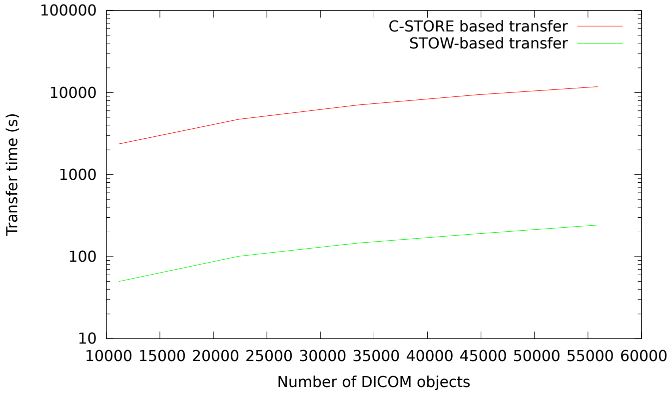

Fig. 4. Time of transfer

performances of STOW are better than C-STORE. On average, STOW's transfer time is 35% lower than C-STORE's. The improvement is higher for the two CT data sets (mean of 51%) than for the single DICOM-MPEG scan (4%). A larger difference is observed for the compressed data set (73%) set than for the un-compressed one (28%). These differences can be explained by the fact that DCMTK's C-STORE waits for an acknowledgment for each transferred object whereas STOW only sends one response at the end of the transfer. The small improvement for the DICOM-MPEG scan is due to the fact that the negotiation time is low with respect to transfer time (data set of $1.1GB$). The greater improvement for the compressed CT data set when compared to the un-compressed one can be explained by the fact that the negotiation time is larger with respect to transfer time when files are smaller.

Experiment 2: transferring functional MRI data sets of varying size: Figure 4 illustrates the transfer time of the MRI data set as a function of the number of files transferred. From the figure it can be noted that STOW is significantly faster than C-STORE (about 50 times on average). We studied the relative and the absolute improvement of STOW as compared to C-STORE. The relative improvement is defined as the difference between both transfer time divided by C-STORE's transfer time. The absolute improvement is the difference (in seconds) of both transfer times. The relative improvement does not depend on the number of files transferred (on average, STOW's transfer time is $98 \pm 0.5\%$ lower than C-STORE's). However the absolute improvement is linear as a function of number of files.

These results can be modeled by a simple formula:

$$t_{C-STORE} = N.(t_f + t_n)$$
$$t_{STOW} = t_r + N.t_f$$

with N the number of files transferred, t_f the transfer time of one file, t_n the C-STORE response time and t_r the HTTP request/response time.

In our experiments, t_r is negligible with respect to $N.t_f$. Thus, the ratio $t_f/(t_f+t_n)$ between transfer times of STOW and C-STORE is constant when the number of files increases. The absolute time difference is linear as a function of the number of files. When larger files are transferred, t_n is negligible with respect to t_f. Thus, the ratio between $t_{C-STORE}$ and t_{STOW} is close to 1. As a result, using the STOW protocol is especially attractive for data set containing a very large number of small DICOM objects (e.g. functional MRI data sets).

V. CONCLUSIONS

In this paper, we compared the DICOM C-STORE to the DICOM STOW protocol. Experiments have revealed that the STOW protocol is faster than C-STORE, especially when transferring a large number of DICOM objects. In practice, the transfer time can be divided by up to 50. While C-STORE has proven its reliability over the past decades, the HTTP-based protocol constitutes a very attractive solution because of its simplicity, ubiquity and its transparent deployment.

REFERENCES

[1] B. Gibaud, G. Kassel, M. Dojat, B. Batrancourt, F. Michel, A. Gaignard, and J. Montagnat, "Neurolog: sharing neuroimaging data using an ontology-based federated approach," in *AMIA Annual Symposium Proceedings*, vol. 2011, p. 472, American Medical Informatics Association, 2011.

[2] IHE International, "Cross-enterprise Document Sharing for Imaging (XDS-I.b) Integration Profile." Integrating the Healthcare Enterprise (IHE), Technical Framework Supplement, March 2009.

[3] W. R. Saywell, "After picture archiving and communication systems: information technology in radiology," *The British Journal of Radiology*, vol. 83, pp. 365–368, May 2010. PMID: 20223902 PMCID: PMC3473577.

[4] O. S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM), A Practical Introduction and Survival Guide*. 2012.

[5] IHE International, "Cross-Community Access for Imaging (XCA-I)." Integrating the Healthcare Enterprise (IHE), Technical Framework Supplement, May 2011.

[6] National Electrical Manufacturers Association (NEMA), Digital Imaging and Communications in Medicine (DICOM) Standard, "Supplement 163: STore Over the Web by REpresentations State Transfer (REST) Services (STOW-RS)," August 2013.

[7] National Electrical Manufacturers Association (NEMA), Digital Imaging and Communications in Medicine (DICOM) Standard, "Part 10: Media Storage and File Format for Media Interchange," 2011.

[8] National Electrical Manufacturers Association (NEMA), Digital Imaging and Communications in Medicine (DICOM) Standard, "Supplement 161: Web Access to DICOM Persistent Objects by RESTful Services (WADO-RS)," February 2013.

[9] "POCO - The C++ Portable Components." http://pocoproject.org/.

[10] OFFIS, "DCMTK - DICOM toolkit." http://dicom.offis.de/dcmtk, 2011.