An Agent-Based Messaging Protocol for Mobile Clinical Communication

Bhuvaneswari Arunachalan and Janet Light, *Member, IEEE*
University of New Brunswick, Canada

***Abstract*—Mobile clinical applications designed for electronic medical record exchange depends on messaging protocols for reliable delivery of critical healthcare data. However, delivery of duplicate messages by these protocols in volatile communication environment often results in irrecoverable medical errors. The methodologies to solve the message duplication problem presented in different areas such as artificial intelligence, network protocols, software architecture, and message specifications address the re-transmission issue by TCP/IP in wired networks. They do not address wireless network issues explicitly. This paper presents a protocol that not only guarantees exact delivery but also checks the validity of messages to avoid duplication and assure exactly once medical record delivery in mobile communication environments.**

***Index Terms*— Electronic Medical Record , Electronic Health Record , Mobile Agent, Migration, Communication Protocol, Health Level 7**

I. Introduction

The increasing awareness of medical errors and the need for better care delivery has raised a demand for electronic access to the health records from anywhere at any time for immediate decision-making and selection of a treatment by both the patients and the healthcare service providers [1][2]. A Canadian study completed in 2004 has identified that between 9,000 and 23,000 Canadians die each year in hospitals from preventable adverse events [3]. This study also shows that 68% of such errors are due to delay in medical diagnosis and its side effects [4][5]. The wide implementation of electronic health record (EHR) systems address this problem by enabling instant access to patients' records electronically. Instant access to EHR requires appropriate data communication system that assures reliability and security of the data exchanged. Different asynchronous communication systems such as electronic mail (e-mail), instant messengers, and synchronous data access systems such as electronic medical record (EMR) systems, clinical application systems are used by healthcare professionals for information exchanges. These systems are implemented in a fixed network environment and operated using desktop computers. With evolving mobile communication technologies, to further increase the efficiency of healthcare services, reliable data communication systems for instant access of EHR using mobile devices will be appropriate and advantageous.

One of the key challenges of mobile clinical systems that enable instant data access to the EHR is the valid and accurate delivery of the EHR content. In general, clinical systems developed for medical data exchange depend on the communication protocols for reliable data delivery. Many clinical systems transmit data using the proficient and widely used TCP/IP protocol for reliable data transmission [22]. However, in a wireless environment, TCP misinterprets interruptions during transmission such as delay (due to link disconnection or signal fading) as congestion. As a congestion control policy, TCP applies retransmission mechanisms to restore the data flow, but often delivers duplicate or multiple copies of a message [23]. Hence, a supportive protocol over TCP/IP is required to prevent message duplication during mobile data transmission.

II. Mobile Clinical Communication

Mobile clinical communication can be specified as an end-to-end data service for inserting, retrieving, modifying, and/or sharing medical data within an EHR system using a mobile device. This service enables real-time exchange of clinical reports for diagnosis and treatment from a remote location by authorized users. The International Standard Organization (ISO) published a technical specification TS18308:2011 [1] for health informatics that defines a set of requirements for the architecture of an EHR system that processes, manages and communicates healthcare information. The specified primary requirements ensure that these EMRs are: clinically valid and reliable, ethically sound, legal, and enable secure data analysis. In 2003, Canada Health Info-way proposed a conceptual EHR solution (EHRS) [2]. The EHRS architecture recommends structured messages for EHR communication that are derived from medical messaging standard such as health level seven (HL7) - clinical document architecture (CDA) specification [24] [30]. The HL7-CDA specification is an internationally recognized standard for reliable and safe exchange of health records. This message specification should be adopted by the messaging protocol used by the mobile clinical applications to achieve global communication. However, these mobile

Manuscript received July 10, 2012. This work was supported in part by the collaborative project carried out by Horizon Health Network, NB and University of New Brunswick, Saint John.

Bhuvaneswari Arunachalan is with the University of New Brunswick, Saint John, NB, Canada E2L4L5 email: a_bhu@yahoo.com.

Janet Light is with the University of New Brunswick, Saint John, NB, Canada E2L4L5 email: jlight@unb.ca.

applications are supported by handheld devices like PDAs, smart phones, and tablet PCs with limited memory resources, high-speed processors, and embedded operating systems.

Mobile clinical applications operate in a network infrastructure identical to traditional wireless network architecture. The difference prevails in the mode of communication and the message delivery model. High-speed wireless links such as CDMA2000, GSM and IEEE 802.11 a/b/g/n can be used for the data transmission [18]-[21]. To prevent data loss and increase throughput, wireless data channels that are augmented with extensive local retransmission mechanisms and channel based scheduling techniques are utilized. For example, the CDMA-1xRTT network applies radio link control (RLC) and radio link protocol (RLP) for reliability. Typical mobile communication architecture is shown in figure 1.

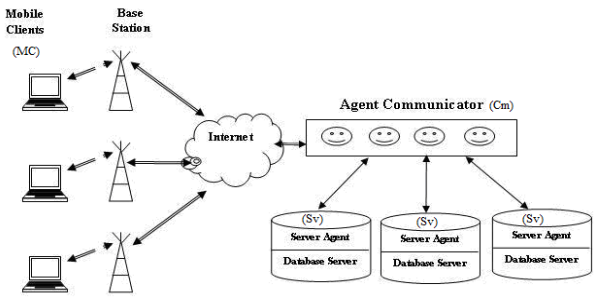

Fig. 1 The mobile clinical communication infrastructure

For simplicity, internals of the sub-networks are ignored and single router network architecture is considered. Here, a mobile client (MC) directly connects with a communicator (Cm) via a wireless link. The mobile link is established through a base station that connects to a Cm using the Internet. The Cm connects to the data server (Sv) through a wired link. The aim is to achieve a guaranteed end-to-end message delivery from the MC to the Sv. The following assumptions are made about the hardware and software involved in this communication process:

- The mobile system is composed of a network of machines and software that are eventually always up and running.
- In case of a delay in acknowledgment message delivery, retransmission of messages is performed until a message is delivered or a session timeout occurs. Here, the mobile agent is assumed to be distorted or failed.
- All transactions between the mobile client and the data server are handled by the communicator. A communicator connects with more than one data server; however, a server connects with only one communicator at an instance.

These assumptions are instrumental in achieving exactly-once delivery of the EMR. The unpredictability of network performance imposes these requirements for additional mechanisms to improve the reliability of this communication.

A great deal of research has gone into implementation of supportive protocols for mobile data access. A significant number of direct data transfer protocols use secured connection-oriented transmission control protocol (TCP). Other widely used protocols are: file transfer protocol, hypertext transfer protocol (HTTP), and simple object access protocol (SOAP) [22][25][26]. They use mechanisms based on store-and-forward or indirect techniques, such as short message service (SMS), multimedia message service (MMS) and electronic mail, for message delivery [27]. Other distributed computing solutions include common object request broker architecture (CORBA), remote procedure call (RPC), Java remote method invocation (RMI), and distributed component objects (DCOM) [28][29]. These solutions are used to support clinical communication through wired networks; yet, a connection component for mobile clinical communication is still to be released.

Existing data access controls such as Active-X data objects, open database connectivity (ODBC), XML database, object linking and embedding (OLE) database, and Java database connectivity (JDBC) provide only moderate support for mobile data transactions [23]. These controls are designed for traditional client-server systems that are inconsistent in mobile environment. EHRi architecture adopts a distributed client/server messaging model for EHR interaction. In practice, the traditional client-server model is no longer valid in the new mobile computing paradigm. This is mainly due to the inadequate functionality of the server computing interface that cannot interact with a mobile client without reloading and processing the code for each transaction. Clearly, this limits the flexibility with which a mobile client can use an existing data server. Various messaging models like an extended client/server model, a remote procedure call (RPC) model, and a mobile agent model are proposed to meet the requirements of mobile communication.

A mobile agent can be defined as an entire computational entity with its code, state, and the resources required for performing the assigned task in a remote node [9]. In a mobile agent system, a piece of code 'B_c' moves with data 'B_d' and state 'B_s' from a client to a server for processing. This process of moving the code from one node to another is called migration. The mobile agents migrate from the mobile client to the EHR data server via wireless networks connectivity. At the server, the agent code is executed when all required credentials are valid. The mobile agent is assigned an itinerary with one node to visit; when the first job is completed, the agent migrates to the data server. Upon completion, the acknowledgment receipt is sent back to the client. The mobile agent paradigm is considered for mobile clinical communication because of its remote execution advantages over the traditional client/server model.

Mobile agent systems often use TCP/IP for transmission of agent code, data and its execution state. However, to perform mobile data transfer, TCP requires external support mechanisms like RPC, object serialization, and data reflection [6]. A better approach is to use a middleware that applies an application layer protocol with appropriate mechanisms above TCP to achieve the required results. This approach is taken in various middleware designed for clinical communication. For example, Aglet [7] uses agent transfer protocol, an application-

level protocol that communicates via a TCP socket; Concordia [8] also uses TCP sockets and Java object serialization as data transfer mechanisms; Odyssey [9] isolates the transport layer from the rest of the system and uses IIOP for communication; Voyager [10] uses Java object serialization and reflection methods extensively in its transport mechanism; Mole [11] initiates a replacement for the Java code loader model by proposing a code server for communication. Though these protocols are reusable, they lack interface specifications to exchange standard messages.

III. Mobile Agent Migration

Agent migration is a process of moving data along with code from one node to another for execution. A migration process is shown in figure 2. In this process, a network model with 1……n nodes is considered. During migration, a mobile agent starts at the source N_i (mobile client) and stops at the destination N_n (server). In order to migrate, the mobile agent needs a reliable migration protocol. The protocol defines the path to travel and the action to perform in each foreign agency that is visited. The sequence of nodes, N_i ($1<i<n$), between N_1 and N_n is called agent itinerary.

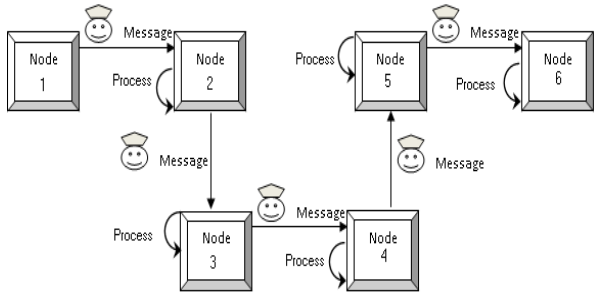

Fig. 2 A mobile agent migration process

A. Life Cycle of the Mobile Agent

A mobile agent is executed only in specified stops, called stages S_i, where i=1…n. A mobile agent at node N_{i-1}, in the corresponding stage S_i, executes the assigned code and moves on to execute at node N_i. The execution of the life-cycle stages results in ordered delivery of an EMR. The life-cycle stages are shown in figure 3. The life cycle of a mobile agent consists of the following five states:

1. In READY state, an agent is created with the EMR as the data content along with the application state and the message delivery code.
2. In MOVE state, the agent migrates between two nodes within the distributed clinical systems network.
3. In RUN state, the agent code assigned to the current state is executed. A copy of the agent is stored locally before execution.
4. In WAIT state, the agent is suspended from its action and the corresponding thread goes to sleep mode. When the execution environment changes, the agent resumes from this state and moves to the RUN state.
5. When the agent completes its assigned task, it reaches the TERMINATE state.

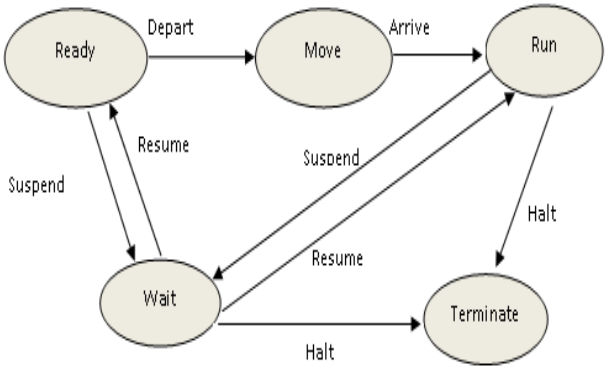

Fig. 3 Life cycle of a mobile agent

Currently, migration mechanisms used in mobile middleware have little or no adaptation to mobile settings. While the problem of guaranteed message delivery is seldom acknowledged, an asynchronous migration mechanism for mobile agents depends on the intelligence of application layer protocols. Depending upon its environment, an agent can enter into any state defined in the model. A mobile agent goes through a life cycle only once.

B. *Migration Mechanisms*

During agent migration, a mobile agent's state, authority, security credentials, data and code are transferred from the mobile client to the communicator (Cm). From agent transfer schemes proposed, three migration mechanisms were identified from Hohl et al. work in [12].

Mechanism 1: In the first scheme, the mobile client pushes all state B_s, code B_c and data B_d of a mobile agent to the Cm in a single instance. In case of an EHR with voluminous content, the migration process consumes all available bandwidth for direct transmission. This scheme is shown in figure 4. This method, in some cases, might utilize additional bandwidth, since Cm might have cached some of the B_c blocks. With suitable mechanisms, re-transmission of these blocks can be prevented.

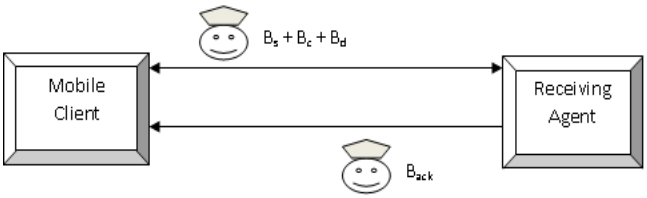

Fig. 4 Migration in a single instance

Mechanism 2: In the second scheme, shown in figure 5, only the mobile agent's B_s, and B_d are transferred in an instance. The Cm requests the client to transmit the missing code blocks only on demand. This scheme reduces bandwidth utilization by transferring code blocks only when required. However,

frequent message exchanges occur through connected links until the end of a session is reached.

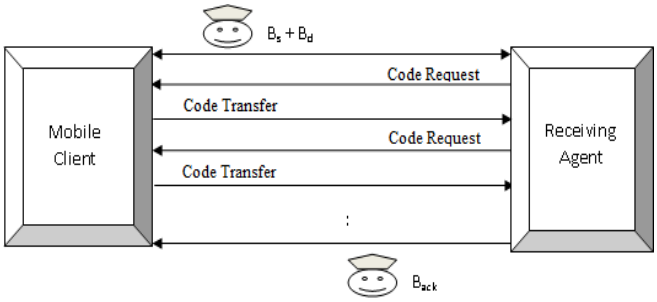

Fig. 5 Migration on demand request

Mechanism 3: The third scheme is a hybrid of the first two schemes (figure 6). Here, a list of B_c blocks needed to perform specific operations are transmitted in an instance. Using this list, the Cm can request required B_c blocks not yet cached. Programming languages such as JAVA and C# provide functions to dynamically load code blocks into an agent location at runtime. This scheme is efficient in terms of resource utilization. However, it is impossible for the Cm to know that the B_c blocks are needed by a mobile agent at any instance.

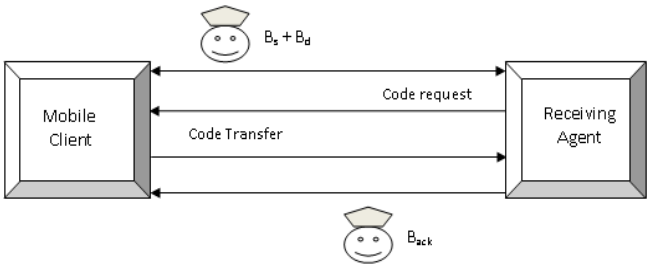

Fig. 6 Migration on a request from the receiver

To simplify the migration process, mechanism 1 is used for the proposed protocol implementation with appropriate mechanisms. The agent migrates between distributed components that are considered as agencies [13]. An agency consists of an associated execution environment where the agents can migrate directly. After successful migration, the code to transfer the data to the appropriate buffer stack is processed. This transfer method is referred to as message delivery. The protocol developed to perform this process is presented in the following section.

IV. The Agent Migration Protocol

The agent migration protocol (AMP) is designed to provide exactly-once execution of a mobile agent with three supporting mechanisms: conformance checking mechanism, message logging mechanism and exactly-once message delivery mechanism. The health domain-specific knowledge is incorporated into the protocol mechanisms to assure maintenance of privacy and confidentiality policy, presented in [14][15]. The agent processing function eliminates message conversions using lower-level protocols. The processes involved in the protocol are shown in figure 4.

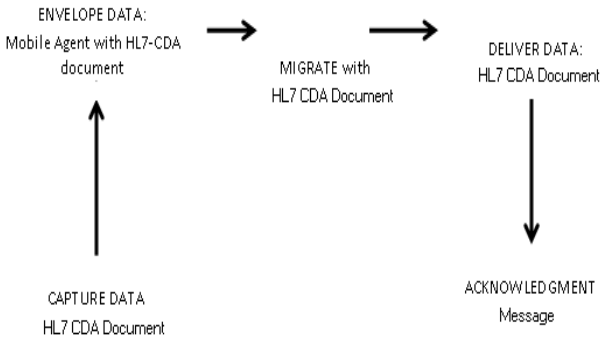

Fig. 7 The process flow of a migration protocol

The protocol provides the following functionalities:

1. The protocol provides functional ability to control messages. At present, a message transmitted from a client is processed by a number of protocol layers that exist between the application and the physical network layers. These layers add redundant headers to the data in the packet before it reaches the receiving application. This methodology introduces inconsistency in message content. To reduce this stress, AMP transfers and receives the clinical messages directly using autonomous objects that are executable only at the designated environment.
2. The protocol integrates a conformance checking mechanism for consistent performance. This mechanism ensures message compatibility between communicating systems, thus eliminating message failure and data loss.
3. In loosely-coupled systems, message latency and throughput determines the performance and scalability of the system. In AMP, the execution time of the agent migration process determines the RTT of a message delivery process. To achieve fast migration, the agent data is delivered as a file (in a batch) and not as a sequence of bytes delivered serially byte by byte.

With these specific functionalities, the AMP presents an asynchronous mechanism as shown in figure 8. The steps of the protocol are presented below.

Step 1: The first step in this protocol is to conform message compatibility between end nodes. The mobile client sends a query agent with a blank message template to the Cm. When the message is received, the Cm creates a checkpoint as described in [16] and forwards the query to the Sv. After receiving the query, a response message with a corresponding message template is send to the Cm by the Sv. The Cm performs the conformance check from the participating nodes. If the templates are compatible, the migration response message is sent to the mobile client. If the templates do not match, then the checkpoint is cleared and a registration failure response message is sent to the client.

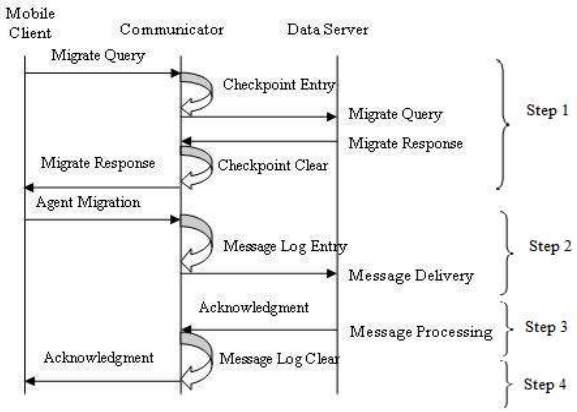

Fig. 8 The agent migration protocol

Step 2: The second step in the protocol is the migration of mobile agents. When a response message is received, the mobile agent migrates from the mobile client to the Cm. After successful migration, a message log is created at the Cm for the mobile agent. A message log is a mechanism used for monitoring the transactions at a node in the distributed system environment. After log creation, the mobile agent is then moved from the Cm to the Sv for message delivery.

Step 3: The third step of the protocol is the clinical message delivery. At the Sv, the agent code is executed using message delivery mechanism for data extraction. The extracted data file is subjected to a verification process. If the file passes the test, a pass code is assigned and the Sv generates an acknowledgment message. A mobile agent with the acknowledgment message is sent to the Cm by the Sv. The Cm verifies the pass code of the agent, clears the message log entry, and moves the mobile agent to the mobile client.

Step 4: The last step in the protocol is the delivery of the acknowledgment message. The client component keeps the agent thread alive until an acknowledgment message is received. Upon delivery of the receipt, the mobile agent reaches its final stage in its lifecycle and terminates its process. The thread is stopped and the code is terminated.

A. Supporting functions

Apart from the above mechanisms, there are two main functions in the protocol: sleep loop procedure and compatibility check procedure. These functions are simple; however, they provide a valuable support to the system.

- *Sleep Loop Procedure:* In case of wait time, the processes call the sleep loop until the state changes. A sleep loop procedure executes a process in a specified interval of time. An interval of 2 seconds is set for each sleep time. The maximum number of times a process can execute the procedure is set to a count of five. This procedure is mainly executed by the Cm while pushing a response message to the mobile client.
- *Compatibility Check:* When a message fails in the conformance checking, it is impossible to establish a successful communication. A proposed solution to this issue is to reconstruct the message according to the server template. The availability of such a mechanism can be indicated in the application profile using a compatibility attribute. If the attribute is set to 'YES', it means there exists a message restructuring procedure. However, this mechanism is implemented here by manual re-transmission.

This protocol has been formalized for use at the application layer of the network protocol suite. The implementation of this protocol is a challenging task that was carried out in the .NET platform. The mobile agent was generated by code written in C# and VB.Net. The performance evaluation of this protocol requires the virtual machine provided by .Net framework to execute the common language runtime (CLR) code be installed in all the communicating nodes.

V. PERFORMANCE EVALUATION

The performance evaluation of the proposed protocol was conducted through a set of experiments in real-world wireless networks. These experiments were conducted using two DELL desktops machines to run the communicator (Cv) and the data server (Sv), and one dv6000 laptop machine to run as the mobile client (MC). The desktops had Windows XP and the laptop runs on Windows Vista operating system. The protocol component is loaded in all of the three machines. In order to evaluate the performance of AMP, experiments were conducted in four different real-world networks shown in table 1. The test environment in a real world wireless network used for performance evaluation is shown in figure 9.

TABLE 1
SPECIFICATION OF THE WIRELESS NETWORKS

Type	Location of mobile client	Speed (Mbps)	Traffic (Users)
A	Campus Network	24	500 – 1000
B	Home Network	1	2
C	Organization Network	10	100 – 200
D	Public Network	36	25 – 50

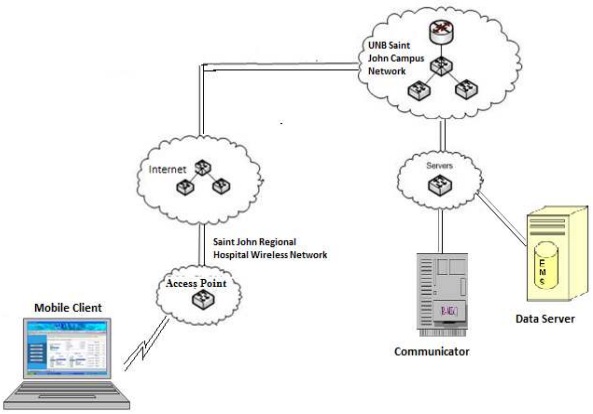

Fig. 9 A real-world AMP test setup

Each network may carry cross traffic from other mobile nodes that might interfere with the packets of the experimental mobile client. This background traffic is vulnerable and sensitive, which is determined by the active users, the applications, and the network characteristics. However, there

are no known techniques to accurately measure the background traffic at successive links in large networks.

In real-time experiments, assumptions were made on the available bandwidth in the networks to be between 1 and 36 Mbps, with network traffic load varying between 10 – 1000 users at a time. The message delivery metrics, the RTT, and the throughput, are measured using DateTime( ) function to evaluate the operational quality using the AMP protocol. The results obtained from these experiments are evaluated using TCP/IP to measure the reliability characteristics: non-duplication and message conformance of the protocol. The TCP/IP measurements are obtained from the testing conducted from the SSL transmission mechanism presented in [17].

The test procedure starts by sending a message from the mobile client to the Sv using the mobile agent in a network (A, B, C, or D). The mobile agent is a two-hop agent; i.e., first, it moves from the mobile client to the Cm; second, the move is from the Cm to the Sv. Each agent starts and returns back to its starting node. This functionality helps in measuring the RTT of the transmission. Two categories of messages, HOT and COLD, are used for this testing. Table 2 shows the message types and their approximate sizes.

TABLE II
TYPE OF MESSAGES USED IN AMP PERFORMANCE EVALUATION

Type of Message	Template Size (KB)	Content Size (KB)
HOT	5	1 – 30
COLD	5	5 – 60

Considering that the data channel established between the mobile client and the base station is subject to delay due to the cross traffic in the network, this network traffic may or may not interfere with the AMP packets. However, this network traffic might change over time and will introduce propagation delay and packet loss between the mobile client and the communicator. The occurrences of cross traffic in a network are unpredictable, as well as the transmission latency.

VI. RESULTS

The measurements presented here are an average of 20 runs, with 7 different message sizes, tested in all four real-world networks. The RTT is deduced from the propagation delay measured in the experiments. The results in figure 10 show the AMP performance in all four networks with different message sizes ranging between 5 KB and 30 KB. As can be seen, migration time is positively correlated with the message size, and it assure 95% confidence that measured migration times are consistent for the messages of different sizes in all four networks. However, the migration time measured in the high-speed networks (network-A and network-D) is lesser than the migration time in the slow-speed networks (network-B and network-C). This value changes as the system configuration and the network infrastructure changes.

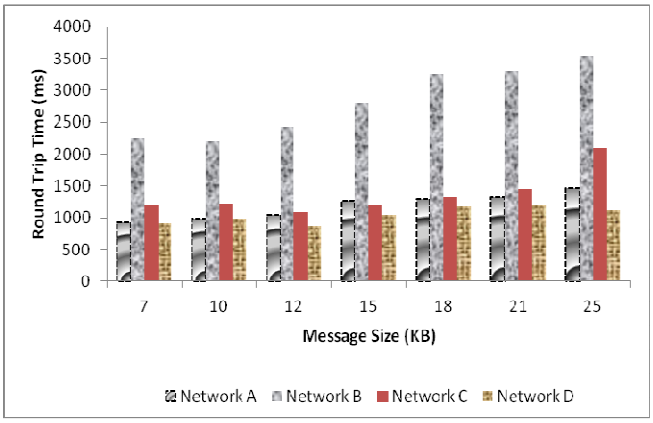

Figure 10 Performance of AMP in different networks

The best migration performance was achieved using network-D, where the smallest agent (>5KB and < 10KB) only needed 966ms for a two-hop migration. The migration time only increased slightly to 1108ms for the largest agent (>30KB and <35KB). Migration in network-A and network-C are only a few milliseconds slower: 966ms and 1228ms for the smallest agent and 1603ms and 1899ms for the largest agent respectively. In network-B, migration was noticeably slower than those using other network types. The migration time taken is 2368ms for the smallest agent and 3462ms for the largest agent. In this process, routing data packets through an access point connected to a heavily loaded router with minimum bandwidth introduces delay in message delivery.

Further, to evaluate the performance of AMP during different clinical events, 7 HOT messages and 7 COLD messages of different sizes are transmitted for 20 times using network-A environment and the migration time or RTT is measured. The average RTT taken for HOT and COLD message transmissions using AMP is shown in figure 11. The results show that the emergency HOT messages are delivered more quickly than the detailed COLD messages. One reason might be the message size, as HOT messages carry filtered dataset. The other reason might be the processing time taken for conformance checking of the message. The HOT messages require less procedure to process the set of elements than the COLD message.

However, an increase in RTT as the message size increases can be observed in figure 11. This overhead is due to the network load imposed by the mobile agent. The network load is imposed by the agent because it not only carries EMR, but also carries associated execution state and code. Also, the agent returns with an acknowledgment receipt for the EMR delivered at the server.

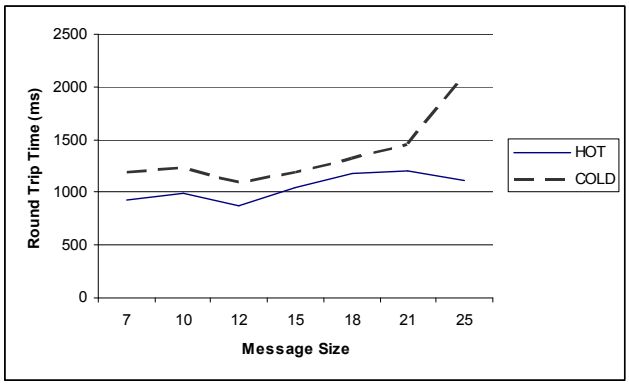

Fig. 11 Round trip time of HOT and COLD messages

The acknowledgment message size remains constant irrespective of the message type. Hence, the time taken for the return trip is independent of the message size. In this process, out of 480 messages (both HOT and COLD) transmitted, only one message failure occurred. This result shows that there exist only less than 0.1% failure rate while using the AMP protocol. However, the failure rate is highly dependent on the transmission media that is extremely volatile in nature.

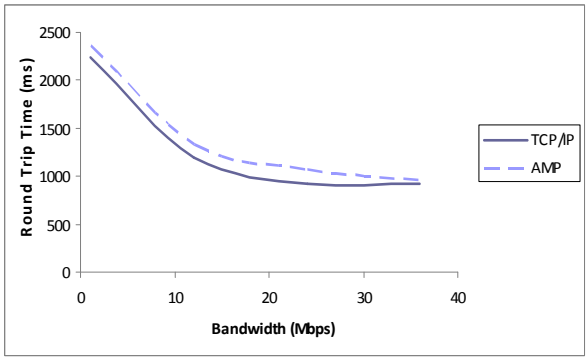

Fig. 12 Round trip time taken by TCP/IP and AMP

With respect to the data transfer rate, significant variation in RTT is observed. The RTT of AMP includes the process time taken for conformance checking and agent execution prior to message delivery, and the delay time during migration from mobile client to the communicator and to the server. To measure the reliability of AMP provided through the conformance check and the exactly-once EMR delivery mechanisms, the performance of AMP is compared with the performance of TCP in networks A, B, C, and D with different data transmission rate. The results are shown in figure 12.

The widely known fact that RTT increases as the available bandwidth decreases is shown in this result. The network D with higher transmission rate shows lower latency during agent migration than other networks. However, the performance of AMP is consistent in all four networks, which suggest that the mechanisms used for conformance checking and exactly-once message delivery are efficient and reliable.

VII. Conclusion

The AMP protocol was developed for EHR communication using mobile agent technology. The results show that the protocol guarantees delivery of clinical records accurately in a wireless environment. This guarantee is ensured through the conformance check on the message prior to data transmission, and the execution property of the mobile agent. Through mobile agents, the messages are validated and delivered exactly-once at the receiver. Also, the protocol handles issues such as delay and data loss caused by the wireless network infrastructure. The performance results are compared with TCP/IP performance to show the significance of AMP with additional mechanisms. Also, the result shows that AMP is consistent and efficient in all of the network environments irrespective of the background traffic and achieves its goal irrespective of the type of messages.

References

[1] ISO18308:2011, "Requirements for an electronic health record architecture",[Online]. Available:http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=52823.

[2] Canada Health Infoway (CHI), "EHRS Blueprint: an interoperable EHR framework". [Online]. Available: http://www2.infoway-inforoute.ca/Documents/EHRS-Blueprint-v2-Exec-Overview.pdf .

[3] Canada Health Infoway, "A Special report on Electronic Health Records for Canadians", [Online]. Available: http://v1.theglobeandmail.com/partners/free/infoway/article canadians.html.

[4] G.R. Baker et al, *The Canadian Adverse Events Study: the incidence of adverse events among hospital patients in Canada,* The Canadian Medical Association Journal, 2004, vol. 170, pp.1678-1686.

[5] T.A Brennan et al, *Incidence of adverse events and negligence in hospitalized patients. Results of the Harvard Medical Practice Study*, The New England Journal of Medicine, 1991, vol. 324, pp.370-376.

[6] D.A. Henderson and T.H. Myer, *Issues in message technology,* In Proceedings of the Fifth Symposium on Data Communications, 1977, pp. 6.1-6.9.

[7] D. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets*, Book, Addison Wesley, 1998.

[8] Mitsubishi Electric ITA, "Technology at a glance: Concordia - Java mobile agent technology", Horizon systems laboratory, 2003. [Online]. Available:http://www.merl.com/projects/concordia/WWW/Concordia-at-a-glance.html.

[9] Brian D.Noble, *Mobile Data Access*, PhD Thesis, School of Computer Science, Carnegie Mellon University, CMU-CS-98-118, 1998.

[10] G. Glass, *ObjectSpace Voyager Core package technical overview*, Book titled: Mobility: Processes, Computers, and Agents, ACM Press/Addison-Wesley Publishing Co., New York, 1999, pp. 611-627.

[11] J. Baumann et al, *Mole 3.0: a middleware for Java-based mobile software agents*, In the proceedings of the IFIP international Conference on Distributed Systems Platforms and Open Distributed Processing, 1998, Springer-Verlag, London, pp. 355-370.

[12] F. Hohl, P. Klar, and J. Baumann, *Efficient code migration for modular mobile agents,* In Proceedings for the Third ECOOP Workshop on Mobile Object Systems, 1997, pp. 1-6.

[13] Foundation of Intelligent Physical Agents, "FIPA ACL Message Structure Specification", *The Foundation of Intelligent Physical Agents*, SC00061G, 2002. [Online]. Available: http://www.fipa.org/specs/fipa00061/SC00061G.html.

[14] J. Light and B. Arunachalan, *Mobile middleware service architecture for EMS application*, Proceedings of 1st ACM-IEEE International Conference on Communication System Software and Middleware, 2006, pp. 1-5.

[15] B. Arunachalan and J. Light, *Middleware service architecture over cellular network for mobile medical applications*, International Journal

of Healthcare Technology and Management, 2007, Vol. 8, No.1/2 pp. 107-119.
[16] Ravi Prakash and Mukesh Singhal, *Low-Cost Checkpointing and Failure Recovery in Mobile Computing Systems,* IEEE Transactions on Parallel and Distributed Systems, 1996, v.7 n.10, p.1035-1048.
[17] J. Light and O.K. Ikejiani, *An efficient wireless communication protocol for secured transmission of content-sensitive multimedia data*, In Proceedings of 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2009, pp. 1-6.
[18] Lawrence Harte, *Introduction to CDMA - Network, Services, Technologies, and Operation*, Book, Althos, 2004.
[19] Groupe Speciale Mobile: GSM World, "History of GSM", [Online]. Available: http://www.gsmworld.com/about-us/history.htm
[20] C. Lindemann, A. Thümmler, *Performance analysis of the general packet radio service*, International Journal of Computer and Telecommunications Networking, 2003, vol. 41:1, pp.1-17.
[21] IEEE 802.11 Wireless LAN: Working Group, "Policies and Procedures", Available: http://www.ieee802.org/11/Rules/rules.shtml
[22] W. Richard Stevens, *TCP/IP Illustrated: The Protocols*, Book, Addison-Wesley Professional Computing Series, 1994.
[23] D. Duchamp, *Issues in Wireless Mobile Computing*, In Proceedings of Third Workshop on Workstation Operating Systems, 1992, pp. 2–10.
[24] Health Level 7 Organization, "V3 Messaging Standard". [Online]. Available: www.hl7.org
[25] Request for comments: 2616, "Hypertext Transfer Protocol – HTTP/1.1", [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt [Accessed: October 21, 2011].
[26] W3C, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", [Online]. Available: http://www.w3.org/TR/soap12-part1/ [Accessed: October 21, 2011].
[27] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services: SMS, EMS and MMS,* Book, John Wiley & Sons, 2005.
[28] Steffen Rothkugel and Peter Sturm, *A CORBA, Java, C++ and ODBMS for Distributed Web Computing - The Taming of the Shrew*, Architektur von Rechensystemen, GI/ITG Fachtagung, 1999, pp. 91-98.
[29] G. S. Raj, *A detailed comparison of CORBA, DCOM, and Java/RMI*, Object Management Group (OMG) whitepaper, 1998.
[30] R.H. Dolin et al., HL7 Clinical Document Architecture, Release 2.0, Journal of American Medical Informatics Association, 2006, vol. 13:1, pp. 30 -39.