Wireless Reliable Messaging Protocol for Web Services (WS –WRM)

Sangmi Lee, Geoffrey Fox
Community Grid Laboratory, Indiana University
{leesangm, gcf}@indiana.edu

Abstract
By employing Web services technology, the Grid system is evolving to be more manageable service infrastructure, including lifetime management, discovery of characteristics, and notifications. Reliable messaging is one of the key issues addressed for quality of services in Web services. In this paper, we propose a reliable message scheme designed for mobile environments in the context of a Web services architecture. We also consider the federation issues with emerging specifications proposed by leading Web services standard groups, Web services –Wireless Reliable Messaging(WS-WRM). We eventually intend to extend the reliability to mobile end-nodes in a more efficient way. We address the design issues and describe the detailed scheme of messaging architecture.

Keywords Web services, Mobile Web services, Reliable messaging, Grid systems, Mobile Grids

1 Introduction

Grid systems provide an environment for sharing heterogeneous resources such as geographically distributed data resources, computing facilities, and scattered sensors. Another emerging technology, Web services architecture, provides an infrastructure for addressing issues related to the discovery and subsequent invocation of distributed services. By aligning with Web services technologies, Grid systems achieve several desirable Web services properties, including service description, discovery, and compatibility with emerging open standards [1]. This in turn ensures that the Grid can be deployed as an environment for sharing and coordinating diverse resources from geographically and organizationally distributed components.

In parallel, there has been an explosive growth of mobile telecommunications networks over recent several years. Just as these mobile communications have changed the accesses to services, it has also influenced the very design of these services. With mobility, the service adapts to new types of service components, and consumers, which work regardless of the location or time. Web services enable mobile devices to actually consume and provide services. There have been efforts to adapt the mobile computing into Web services architecture [2 ~ 4].

The CAROUSEL project from Community Grid Laboratory has investigated adapting mobile devices into the collaborative environment within Web services architecture [5]. In CAROUSEL Web service, mobile devices are adapted with the user input/output defined intelligently by interaction between the user "profile" and the semantics of general Web services [6], [7]. The NaradaBrokering system [8] from Community Grid Laboratory provides the core event brokering service for the CAROUSEL project. NaradaBrokering is a general purpose event brokering system designed to run a large network of cooperating broker nodes. NaradaBrokering system efficiently routes any given event between the originators and registered consumers of the event in question. Communication within NaradaBrokering is asynchronous and the system can be used to support different interactions by encapsulating them in specialized events. With these event encapsulations, NaradaBrokering successfully supports higher level of architectures such as JMS (Java Message Service) [9] based on a message oriented model, or Web service architecture [10] implementing the service oriented model.

Meanwhile, as interaction between Web services are increasingly dynamic and sophisticated, the leading groups within the Web services community are suggesting solutions for reliable message delivery[11], [12]. Considering the higher failure rates prevalent in mobile computing, it is even more important in mobile Web services. However, since current mobile networks are considerably different in its underlying communication services from fixed networks', it is hard to adapt the aforementioned suggested specifications without modifications. Here are the limited capabilities of current 3G-based mobile networks which should be considered.

- *Limited underlying network protocols*. In contrast to fixed network communications, most of the current 3G based mobile services provide limited protocols in their data communication network.

Therefore, even in the case of developing based on the proxy architecture, most of the current mobile devices are restricted to using the HTTP protocol, which is a request-response messaging scheme different from the bidirectional dynamic communications over a raw socket connection. This limitation turns out to be a critical obstacle for improving performance in the proxy-based architecture.

- *Communication handovers in the service Gateway.* 3G based data communication services are performed via the communication gateway of the wireless telecommunication service vendors. The service gateway processes the handover between wireless telecommunications and the Internet, so that the mobile users can access the resources sitting on the edge of fixed networks. Therefore, the mobile application is not independent of the wireless telecommunication providers.
- *Limited local storage in mobile devices.* Compared to fixed network nodes, mobile devices do not have sufficient data storage. Hence, there is a limit on the member of messages that can be stored locally for the purpose of failure recovery.

However, current major specifications, including WS-Reliability [11] and WS-ReliableMessaging [12], are designed based on services running on a fixed network, without considering the limitations mentioned above. Both of schemes are based on ACK-based reliability schemes which need additional number of underlying network requests to send the acknowledgements. Obviously, this can cause visible time delays in a given mobile Web service which traditionally has approximately 10,000 times lower bandwidth than LAN based services.

Therefore, we propose a reliable messaging scheme for mobile Web services which is optimized for the mobile network environment and compatible with WS-Reliability and WS-ReliableMessaging: WS-Wireless Reliable Messaging (WS-WRM). The WS-WRM suggests using positive and negative acknowledgement messages, as known as ACK and NAK messages, to optimize the data communication overhead. The four basic delivery assurances are provided in WS-WRM; AtMostOnce, AtLeastOnce, ExactlyOnce, and InOrder.

WS-WRM provides a flexible syntax to interact with WS-Reliability and WS-ReliableMessaging in the NaradaBrokering system. We provide enough flexibility within the scheme, so that the reliable messages from different reliability schemes can be processed in an optimized way for mobile end points, still satisfying initial assurance scheme from end to end.

In addition to the basic types of assurances, we provide a message integrity element. To ensure that the message content is delivered correctly, a sender node can add the message digest and request integrity checking.

This paper is organized as following. In section 2, we describe the related researches. The requirements of design WS-WRM scheme is followed in section 3. In prior to discuss about the reliability scheme in WS-WRM, we described a limited HTTP binding issues in section 4. In section 5, we illustrate the basic scheme and simple example. The future work and conclusions are followed in section 6.

2 Related Work

The application layer reliability is not unique to Web services. There have been reliable messaging schemes in the application layer for XML-based collaborative applications. Among the earlier works, there have been efforts such as RosettaNet PIP model [13] and the BizTalk framework [14] which provide business-level reliable messaging. Both of them exchange positive or negative acknowledgement messages within their business architectures. A more general approach was specified by ebXML with its ebXML Message Service (ebMS) [15] which is the first reliability scheme binding with SOAP message and the antecedent of WS-Reliability. The reliability in ebMS is from the exchange of ebMS Message Service Handler (MSH) responding to a message with an Acknowledgment message.

WS-Reliability [11] has extended and modified the basic ebMS scheme within the Web service domain. WS-Reliability provides reliable message delivery within the Web services context. WS-Reliability defines mechanisom to guarantee message delivery, eliminate duplicate messages, and guarantee received message order. Similarly, the competing specification provides a WS-ReliableMessage [12] reliability scheme for the Web services architecture. The reliable messaging model is pretty identical to that of WS-Reliability. Compared to WS-Reliability, WS-ReliableMessaging is dependent on the WS-Policy to specify the assurance pattern. Pertaining to Quality of Service, Message senders and receivers exchange the policy initially, and perform the assurance process according to the policy.

Assurance scheme is also different. WS-Reliability requires the exchange of the acknowledgement message for every reliable message; meanwhile WS-ReliableMessaging requests the acknowledgement only within the last message in the message group which is delivered sequentially. Even though the WS-ReliableMessaging provides an element which mandates acknowledgements for every message, the basic acknowledgement schemes show differences.

Nevertheless, both the reliability schemes for the Web services architecture require ACK based acknowledgement exchange for assurances. Therefore, it is hard to deploy the solutions for mobile Web services without modifications, since the underlying network has limited bandwidth and the device has limited data storage. WS-WRM employs ACK/NAK-based acknowledgement scheme to optimize the data transactions.

Moreover, we are investigating the federation of heterogeneous reliability schemes on top of the NaradaBrokering system. The end-points can talk to each other with their own reliability scheme - WS-Reliability, WS-ReliableMessaging, and WS-WRM. We develop WS-WRM as one of the reliability scheme supported by the NaradaBrokering system for mobile end points. Most of the intelligent features are designed to be located on the messaging service to optimize the memory usage on the mobile devices.

3 Requirements for design of WS-WRM

Now we have fundamental requirements to extend the reliable messaging for Web services;

- *Reliable messaging delivery assurances*. Mobile end point should be able to provide four basic delivery assurances, AtMostOnce, AtLeastOnce, ExactlyOnce, and InOrder in the Web services domain.
- *Optimization of the acknowledging scheme for better performance*. WS-WRM should provide reliability scheme with less data transmissions, so that a mobile Web service can be orchestrated with services running in fixed-network environments.
- *Optimization of the usage of local data storage in mobile devices*. WS-WRM scheme should factor the limited data storage on the mobile end-points.
- *Stronger delivery assurance*. Mobile network communication has a higher failure rate, hence, reliability scheme requires additional methods to ensure the integrity of the delivered message.
- *Binding with other Web services features*. The schema should be able to bind with other Web services standards, such as SOAP message and WS security scheme.
- *Compatibility to WS-Reliability and WS-ReliableMessaging*. The protocol should provide flexibility in its syntax to process various requirements of the application. WS-WRM also interoperates with the heterogeneous reliability scheme in the NaradaBrokering system. Therefore, reliable messages from WS-Reliability or WS/ReliableMessaging can be delivered to mobile end point reliably, and satisfy the original assurance scheme.
- *Adaptability*. It should be easy to adapt WS-WRM to new devices without loss of reliability.
- *Binding with the event brokering system*. WS-WRM is designed to perform on top of the NaradaBrokering event brokering system. Therefore, it should be optimized for use with the resources provided by NaradaBrokering system, such as stable event storage.

Some of the above requirements are regarding the design of the protocol, and the others are implementation issues. In this paper, we focus more on the design of the protocol in the application layer.

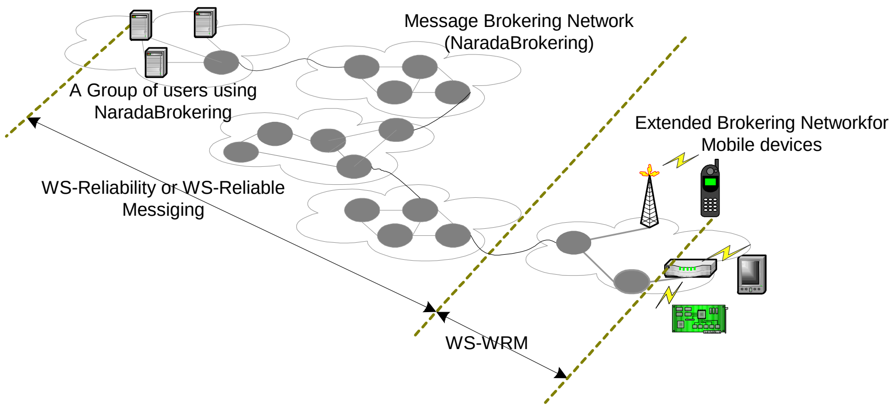


Figure 1 Extending reliable messaging to mobile devices within the Web services architecture

4 WS-WRM and the Limited HTTP binding

The WS-Reliability specification from OASIS presents three patterns of HTTP binding to implement the reliable messaging protocol; Request-Response, Callback, and Polling [11]. However, for mobile devices, not every type is available. Since most of mobile devices are limited to perform as a client of HTTP, the callback messaging type which requires both the sender and receiver being able to initiate requests, is not available. The poll binding pattern is partially available, when the mobile end-node sends a reliable message.

Despite the limited capability in network communications, mobile devices are also required to be clients and service providers in the Web services architecture. For this issue, we classify the approaches into two different types; within the transport layer or SOAP message layer.

The approaches in the transport layer pertain to the messaging architecture complementing the limited underlying network capability. For example, instead of binding to HTTP directly, the protocol can be implemented on top of the NaradaBrokering service. Since the NaradaBrokering mobile service provides connectivity to the event brokering system, the service does not have to consider the limited HTTP service on the mobile device.

The Second approach is specifying a HTTP request in the reliable message within the protocol. Without the specially designed messaging architecture, the HTTP request of the underlying HTTP protocol can be specified within the SOAP message layer. In the context of the SOAP message layer, WS-WRM provides a `<PollingRequest>` element, which specifies next polling within the time specified in the `<PollingExpirytime>` element. Initially WS-WRM sets a default polling expiry time for each mobile device. Therefore, if there is no specification of `<PollingExpirytime>` from the message sender, the mobile devices will poll every time the "polling expiry time" is expired.

Figure 2 illustrates an example of the HTTP binding from the sender to mobile end-node via NaradaBrokering. This example shows that a reliable message from the sender is delivered to the mobile destination with WS-Reliability scheme without modification. To perform a guaranteed message delivery of the WS-Reliability, the actual underlying communication performs additional HTTP request-response ((2), and (6)). In addition to the lower bandwidth of wireless communication, the additional data transmissions can be a critical obstacle to cooperate with fixed-network based services. This example illustrates how significantly the reliability scheme can influence the actual performance in the mobile environment. Here, we need to optimize the acknowledging scheme to account for characteristics inherent in mobile communication.

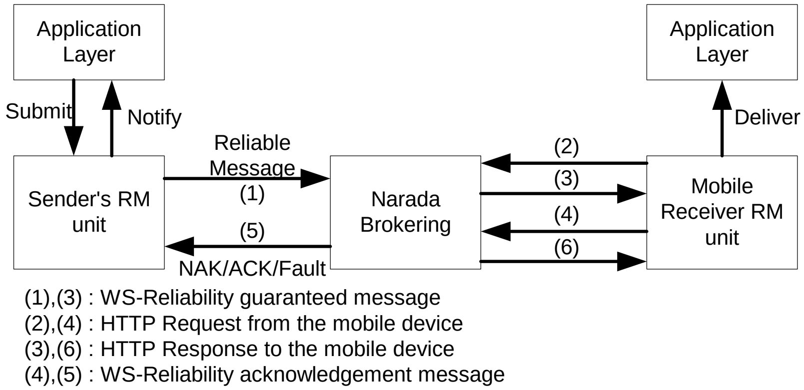


Figure 2 Example of HTTP binding to implement a guaranteed message delivery with WS-Reliability in mobile environments

5 ACK/NAK based Message Delivery Assurances

In the design of a reliability scheme, decision of ACK vs. NAK is one of key issues. Using the ACK based reliability scheme is implementations tend to be simple and clearer: WS-Reliability and WS-RM also follow the ACK based reliability scheme. Nevertheless, we should consider the limited characteristics of current mobile technologies, comparably long latency and limited network protocols. To be orchestrated with other fixed network based Web services, mobile devices should optimize the number of requests which are processed in the underlying network protocol.

WS-WRM is designed as an ACK/NAK-based reliability scheme. The receiver sends NAK message only if there is delivery failure. Figure 3 illustrates an example of message exchange in WS-WRM. The WS-WRM end point includes the mobile end point, proxy server, or messaging service. Every message contains a sequential integer number to track and manage the reliable delivery. Each WRM message must have an integer value of <MessageID> to number the messages started from 1. If the NAK tracking is requested in the <Request> element, the destination tracks the number of messageID, and if the message number is not ordered, the destination will detect the message loss and send a NAK message to sender. Now we describe how the <ACKRequested> element works. An endpoint can only detect losses with the messages that were published prior to the last message that it delivered. Therefore, the sender sends the request of acknowledgement to make sure that the last message is received successfully. The time interval between the requests of acknowledgement depends on the policy between the mobile end-point and proxy. For some extreme cases, such as secure banking transactions, an acknowledgement can be requested for each message.

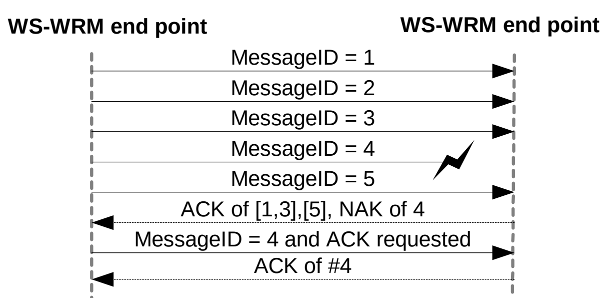


Figure 3 Guaranteed Message Deliveries in WS-WRM

5.1 Message Structure

The reliability in WS-WRM is completely specified on the sender side. There is no policy exchange prior to the data transfer. Every policy assurance policy is performed by specific request and response. Therefore each end point can provide assurance features that the service requires.

We now outline issues that pertain to the adaptability of new emerging mobile devices. First, as a destination of a reliable message, new mobile devices is required to process responses according to the request specified in a reliable message. It does not have to understand policies or perform it intelligently. Therefore, developers can adapt a new device easily. Second, as a sender of a reliable message, new mobile devices can choose only the required assurance type for sending a message. Instead of supporting a fully functional assurance scheme, each device can support a selected assurance scheme only.

Now here we describe the structure of WS-WRM message. There are five elements; <MessageHeader>, <Request>, <Response>, <PollingReqeust>, and <Fault>.

- *<MessageHeader> Element*. The <MessageHeader> element is a required element for every types of message including guaranteed delivery message, non-guaranteed delivery message, response message, or fault message. Every message includes <MessageID> element which is a monotonically increasing integer number. The number starts at 1 and rolls over when it reaches the maximum value of integer. The <ExpiryTime> is included to specify how long the request can wait for the response from the destination. When the <Request> element includes <NAKTrackingRequired> element, the message receiver should keep tracking the sequence of messages with <MessageID> element.
- *<Request> Element.* The <Request> element must be presented in WS-WRM message. The sender specifies assurance in the <Request> element. ACK, NAK tracking, message integrity, duplication elimination, and message delivery in order are specified in this element.
- *<Response> Element.* The Response element includes response information for NAK and ACK message. <Response> element pertains to the response from the destination which performs the assurance scheme specified in the request element. Each NAK and ACK message contains the message number(s) involved.
- *<PollingRequest> Element.* To invoke the HTTP request from mobile devices, WS-WRM provides a polling request. This is an optional element. When the application is not designed on top of an intelligent messaging architecture supporting communication between mobile devices and fixed network based devices, or the device is significantly light client, the sender can specify the polling and the time interval.
- *<Fault> Element.* <Fault> element specifies exceptions that occur during message delivery.

Figure 4 presents how above elements are coordinated together in a reliable message.

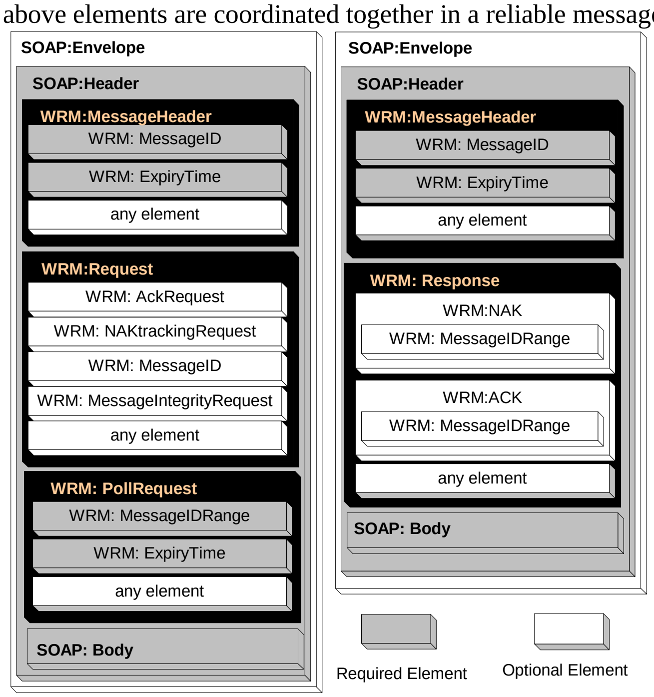


Figure 4 Message Structure of WS-WSM

5.2 Guaranteed Message Delivery

A reliable message can specify <NAKTrackingRequest> element in the <Request> element. If the destination received the NAK tracking request, the destination starts to keep track of the order of message IDs in delivered messages. The message ID is monotonically increased by 1 and rolls over if the number reaches the maximum value of integer. If the destination detects the missing message ID, it sends a NAK request to the sender. The sender sends the message again with an ACK request. This ACK request includes the expiry time, and if there is no response from destination within the expiry time, the sender will send the message again.

The message sender can send ACK request message to destination whenever it requires. ACK request includes message IDs for which the sender has not received acknowledgements. The tradeoffs between the frequency of ACK request and performance are expected.

5.3 Duplication Elimination and Guaranteed message ordering

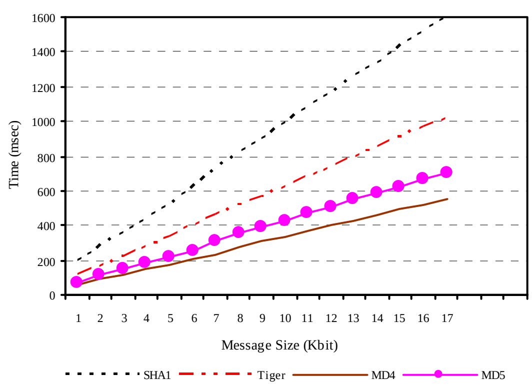
Similar to the guaranteed message delivery, duplication elimination and guaranteed message ordering are also specified in <Request> element. The receiver keeps the id of reliable message received recently, and performs the assurance requests.

5.4 Message Integrity Request

As an additional delivery assurance type, we provide a message integrity feature. If the sender requests message integrity validation with generated message digest, the destination generates the SOAP message body with specified algorithm and verifies the integrity.

Figure 5 shows various performances between the algorithms of generating message digest. WS-WRM provides multiple algorithms, and the services can specify the algorithm based on their requirements. Currently, MD5 algorithm is supported as default. Figure 5 and 6 show the cost of validating the message with message digest. The measurement is performed with,

- Device: PalmOne's Treo300
- Platform: J2ME CLDC 1.0
- Processor Technology: 33 MHz Motorola Dragonball VZ
- Memory: 16 MB
- Operating Systems: Palm OS 3.5.2H



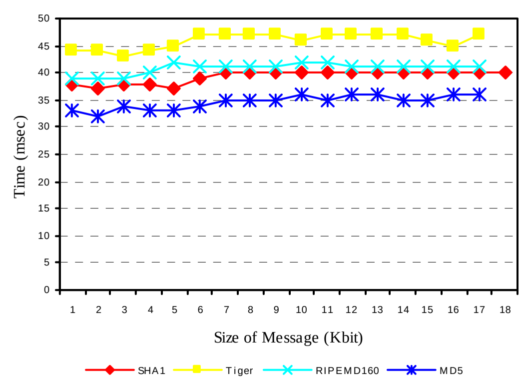


Figure 5 Generating message digest on Treo300 **Figure 6 Comparison of message digests on Treo300**

5.5 Interacting with the heterogeneous reliability schemes

WS-WRM is designed to interoperate with other reliable delivery schemes over the NaradaBrokering system. A message from the original sender goes through two phases; from the original sender to NaradaBrokering and from NaradaBrokering to the mobile device. NaradaBrokering performs assurance

response to each end-device each of which may have a different assurance scheme. Therefore, if there is a message loss between the original sender and NaradaBrokering, NaradaBrokering performs every reliable delivery and fault recovery instead of the mobile end-point. Since NaradaBrokering is curretrn incorporating support for both WS-Reliability and WS-RM, each assurance is supported in this architecture. Figure 7 is an example of exchanging a reliable message between WS-ReliableMessaging sender and WS-WRM destination via NaradaBrokering system. In the example of the figure 7, the original sender intends to send five reliable messages with the assurance type of guaranteed delivery. Second message is lost before reaching to the NaradaBrokering system, and the fourth message is missed between the NaradaBrokering and mobile end-point. Figure 7 shows how these different reliability schemes perform together while satisfying constraints specified in these architectures.
As we illustrate in the example of figure 7, cooperating multiple reliability schemes requires intelligent support from the messaging service. To satisfy different schemes, NaradaBrokering understands each scheme completely and performs the assurance mechanisms, so that the users can have illusion of talking to the others in the same reliable scheme as theirs. The most important requirement is that the merged system should satisfy "end-to-end" reliability in the application layer.

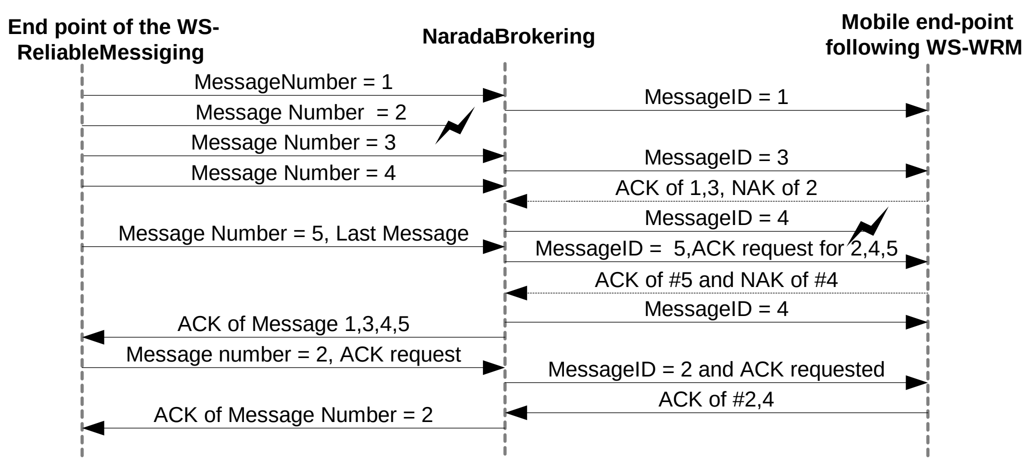


Figure 7 Example of cooperating WS-WRM and WS-ReliableMessaging

6 Conclusions and Future work

Extending the reliable messaging to the mobile end-node is required to adapt mobile devices as a service provider in the context of Web services environments. The message reliability should be guaranteed for complete service functionality. However, there are difficulties in adapting existing or proposed reliability schemes without modification specific to mobile computing, which has higher failure rate, lower bandwidth, and limited data storage. In this paper we discussed issues, and our strategies to extend the reliability efficiently. We proposed WS-WRM, which is a Web services reliable message protocol designed for wireless communication environment. In this paper, WS-WRM is described focused on its protocol in application layer. Implementation of this architecture also has various practical issues that will be discussed in future research papers.
The implementation of WS-WRM also pertains to federating with different reliability schemes especially for the WS-Reliability and WS-ReliableMessaging. The cooperative reliable scheme is designed on top of the NaradaBrokering service. WS-WRM provides flexible syntax to cooperate with different schemes.
Another area of research is the incorporation of other Web services standards. Extending Web service standards regarding security, policy and notification needs to be investigated for mobile environments, so that Grid systems can achieve more complete sharing of the heterogeneous resources in mobile environments.

Reference

[1] S. Tuecke, K. Czajkowske, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, and D. Snelling, Open Grid Services Infrastructure (OGSI) Version 1.0 Global Grid Forum Recommendation, 6,27,2003. Available from http://www.gridforum.org/ogsi-wg

[2] Microsoft Inc., Mobile Web Services Technical Roadmap, 2003, Available from http://www.microsift.com/serviceproviders/mobilewebservice
[3] IBM Inc., Web Services Toolkit for Mobile Devices, 2002, Available from http://www.alphaworks.ibm.com/tech/wstkmd
[4] J. Ellis, M. Young, J2ME Web Services 1.0, Available from http://jcp.org/aboutJava/communityprocess/first/jsr172/index2.html
[5] S. Lee, S. Ko, G. Fox, K. Kim, S. Oh A Web Service Approach to Universal Accessibility in Collaboration Services in Proceedings of 1st International Conference on Web Services Las Vegas June 2003
[6] G. Fox, S. Ko, K. Kim, S. Oh, S. Lee on "Integration of Hand-Held Devices into Collaborative Environments" IC'02 June 2002 Las Vegas, NV
[7] S. Oh, G. Fox, S. Ko GMSME: An Architecture for Heterogeneous Collaboration with Mobile Devices The Fifth IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2003) in Singapore October, 2003.
[8] G. Fox, S. Pallickara, NaradaBrokering: An Event Based Infrastructure for Building Scaleable Durable Peer-to-Peer Grids. Chapter 22of *Grid Computing: Making the Global Infrastructure a Reality Grid.* Published by John Wiley, West Sussex, England. ISBN 0-470-85319-0. 2003.
[9] G. Fox, S. Pallickara, JMS Compliance in the Narada Event Brokering System. Proceedings of the *2002 International Conference on Internet Computing* (IC-02). Volume 2 pp 391-397
[10] H. Bulut, G. Fox, S. Pallickara, A. Uyar and W. Wu Integration of NaradaBrokering and Audio/Video Conferencing as a Web Service. *IASTED International Conference on Communications, Internet, and Information Technology, 2002*. pp 401-406.
[11] K. Iwasa, OASIS Web Services Reliable Messaging TC WS-Reliability, Working Draft 0.91, 2,2,2004, Available from http://www.oasis-open.org/committees/documents.php?wg_abbrev=wsrm
[12] R. Bilorusets, A. Bosworth, D. Box, F. Cabrera, D. Collison, et. el. Specification: Web Services Reliable Messsaing Protocol (WS-ReliableMessaging), 3,13,2003, Available from http://www-106.ibm.com/developerworks/library/ws-rm/
[13] RosettaNet, http://www.rosettanet.org/RosettaNet/Rooms/DisplayPages/LayoutInitial
[14] Microsoft Inc., BizTalk, Available from http://www.microsoft.com/biztalk/techinfo/productdoc/2002/BTdownload.asp
[15] M. MacKenzie, Adobe Systems, I. Jones, BT Group plc., OASOS ebXML Messaging services, 3.0, Available from http://www.oasis-open.org/committees/documents.php?wg_abbrev=ebxml-msg