Network Working Group   J. Wong, Ed.
Request for Comments: 4416   Nortel Networks
Category: Informational   February 2006

Goals for Internet Email to Support Diverse Service Environments

Status of This Memo

Copyright Notice

Abstract

This document is a history capturing the background, motivation and thinking during the LEMONADE definition and design process.

The LEMONADE Working Group -- Internet email to support diverse service environments -- is chartered to provide enhancements to Internet mail to facilitate its use by more diverse clients. In particular, by clients on hosts not only operating in environments with high latency/bandwidth-limited unreliable links but also constrained to limited resources. The enhanced mail must be backwards compatible with existing Internet mail.

The primary motivation for this effort is -- by making Internet mail protocols richer and more adaptable to varied media and environments -- to allow mobile handheld devices tetherless access to Internet mail using only IETF mail protocols.

The requirements for these devices drive a discussion of the possible protocol enhancements needed to support multimedia messaging on limited-capability hosts in diverse service environments. A list of general principles to guide the design of the enhanced messaging protocols is documented. Finally, additional issues of providing seamless service between enhanced Internet mail and the existing separate mobile messaging infrastructure are briefly listed.

Table of Contents

1. Introduction

Historically, a number of separate electronic messaging systems originated and evolved independently supporting different messaging modes. For example:

- o Internet mail systems ([4], [10], [25]) evolved to support networked computers with messages consisting of rich text plus attachments.
- o Voice mail systems utilized a client with a telephone-based or an answering machine style of user interface. The telephone network was used for transport of recorded voice messages.
- o Fax store-and-forward users interface with a fax machine using a modified telephone-based interface. Fax machines use the telephone network for transport of fax data via modems.
- o SMS (Short Message Service) [58] enabled users to send short text messages between their cellular phones using the SS7 call control infrastructure ([60], [61], [63], [64], [65]) for transport.

In the recent past, IETF mail standards have evolved to support additional/merged functionality:

- o With MIME ([5], [6], [7], [8], [9], [28]), Internet mail transport was enhanced to carry any kind of digital data
- o Internet mail protocols were extended and profiled by VPIM ([13], [14], [15], [34]) and iFAX ([16], [17], [18], [19], [20], [21], [23]) so that enabled voice mail systems and fax machines could use the common email infrastructure to carry their messages over the Internet as an alternative to the telephone network. These enhancements were such that the user's experience of reliability, security, and responsiveness was not diminished by transport over the Internet.

These successes -- making Internet mail transport the common infrastructure supporting what were separate messaging universes -- have encouraged a new vision: to provide, over the Internet, a single infrastructure, mailbox, and set of protocols for a user to get, respond to, and manipulate all of his or her messages from a collection of clients with varying capabilities, operating in diverse environments ([46],[47]).

The LEMONADE effort -- Internet email to support diverse service environments -- realizes this vision further by enabling Internet mail support for mobile devices and facilitating its interoperability with the existing mobile messaging universe.

In the recent past, the evolution of messaging standards for resource-limited mobile devices has been rapid:

- o In the cellular space, SMS was enhanced to EMS (Extended Message Service) [59] allowing longer text messages, images, and graphics. With an even richer feature set, MMS (Multimedia Messaging Service) ([43], [52], [53], [56], [57]) was developed as a lightweight access mechanism for the transmission of pictures, audio, and motion pictures. MMS protocols are based in part on Internet standards (both messaging and web [24]) as well as SMS. The cellular messaging universe is a separate infrastructure adapted to deliver appropriate functionality in a timely and effective manner to a special environment.
- o As well, the number of different mobile clients that need to be supported keeps proliferating. (e.g., besides cellular phones there are wireless-enabled PDAs, tablet computers, etc.)

These resource-limited mobile devices are less powerful both in processing speed and display capabilities than conventional computers. They are also connected to the network by wireless links whose bandwidth and reliability are lower, latency is longer, and costs are higher than those of traditional wire-line links, hence the stress on the need to support adaptation to a whole different service environment.

This document collects a number the issues impeding Internet mail protocols from directly supporting the mobile service environment. Considerations arising from these issues are documented, and in some cases possible approaches to solutions are suggested. It turns out that the enhancements to support mobile clients also offer benefits for some terminals in other environments. In particular, the enhancements address the needs of the following diverse clients:

- o A wireless handheld device with an email client -- a Wireless User Interface (WUI) mode of user interaction is dictated by the constraints of the mobile wireless handheld operating environment.
- o Telephone-based voice client -- a Telephone User Interface (TUI), this is the user mode offered by a POTS set
  - * This is a subset of the WUI and is useful in other contexts.
- o A multi-modal messaging client providing a coordinated messaging session using display and audio modes simultaneously. (e.g., a system consisting of a PC with a phone, or a wireless phone with both a voice circuit and data channel requiring coordination).
  - * This is also a subset of the WUI and is useful in other contexts.

The rest of this document is structured as follows:

- o A brief survey of messaging profiles - both existing and proposed.
- o A list of principles to be used to guide the design of Internet Messaging for diverse service environments.
- o Detailed discussion on enhancements to Internet mail protocols to support WUIs.
- o Some issues relating to the interoperation of enhanced Internet mail and the existing mobile messaging services.

2. Conventions Used in This Document

This document refers generically to the sender of a message in the masculine (he/him/his) and to the recipient of the message in the feminine (she/her/hers). This convention is purely for convenience and makes no assumption about the gender of a message sender or recipient.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [1].

3. Messaging Terminology and Simple Model (Client-to-Server Aspect Only)

In the client-server model prevalent in existing messaging architectures, the client, also known as a "user agent", presents messages to and accepts messages from the user. The server, also known as a "relay/server" or a "proxy-relay", provides storage and delivery of messages.

For a definitive description of Internet mail architecture, see [42].

3.1. Messaging Transaction Models

There are two basic transactional models. In the "pull" model, the component, rather than the data flow, initiates the transaction. For example, a client may initiate a connection to a server and issue requests to the server to deliver incoming messages. Conventional email clients, web-mail clients, and WAP-based mobile clients use the "pull" model.

The "push" model differs in that the component initiating the transaction does so because of some data flow affecting it. For example, the arrival of a new message at the terminating server may cause a notification to be sent ("pushed") to a messaging client.

3.2. Mobile Messaging Transactions

The most common functions are: "submission", "notification", and "retrieval". There may be other functions, such as "delivery reports", "read-reply reports", "forwarding", "view mailbox", "store message", etc. Each of these transactions can be implemented in either a pull or push model. However, some transactions are more naturally suited to one model or another.

The following figure depicts a simple client-server model (no server-to-server interactions are shown):

(1) Message submission
(2) Message notification
(3) & (4) Message retrieval

```
      +-------+                 +------+                       +-------+
      |Mail   |-------(1)------>|      |-----------(2)-------->|Mail   |
      |Client |   Submit msg    |      |     Notification     /|Client |
      +-------+                 |      |                     / +--+----+
                                |      |                    /     ^
                                |      |<----------(3)-----+     /
                                |Server|   Retrieval request    /
                                |      |                       /
                                |      |                      /
                                |      |-----------(4)-------+
                                |      |   Retrieval response
                                |      |
                                +------+
```

Simple Messaging Model

3.2.1. Submission

"Submission" is the transaction between a client and a server by which the user of the former sends a new message to another user. Submission is a push from client to server.

3.2.2. Notification

"Notification" is the transaction by which the server notifies the client that it has received messages intended for that client. Notification is a push from server to client.

All the larger mobile messaging systems implement a push model for the notification because data can be presented to the user without the user's experiencing network/transport latencies, and without tying up network resources for polling when there is no new data.

Internet mail differs in that it has not yet seen the need for a standardized notification protocol.

3.2.3. Retrieval

"Retrieval" is the transaction between a client and a server by which the client can obtain one or more messages from the server. Retrieval can be push or pull.

Implemented in some mobile systems as an option, the push model has the advantage that the user is not necessarily aware of transport or network latencies.

The pull model, implemented in most systems (mobile or conventional), has the advantage that the user can control what data is actually sent to and stored by the client.

4. Profiles

Internet messaging can be made to support a variety of client and server types other than traditional email. The clients may be adapted for host restrictions such as limited processing power, message store, display window size, etc. Alternatively, clients may be adapted for different functionality (e.g., voice mail, fax, etc.). Servers may support optional mail features that would allow better handling of different media (e.g., voice mail, fax, video, etc.). A number of Internet mail profiles supporting specific application niches have been defined or proposed.

4.1. Existing Profiles

The following are examples of server-to-server profiles of SMTP and MIME. Except for IVM, they do not address client-to-server interactions.

4.1.1. Voice Messaging (VPIMv2)

These profiles, RFC3801 [13] to RFC3803 [15], enable the transport of voice messages using the Internet mail system. The main driver for this work was support of IP transport for voice mail systems. As voice mail clients are accustomed to a higher degree of responsiveness and certainty as to message delivery, the functionality added by VPIMv2 includes Message Disposition

Notification and Delivery Status Message ([12], [3]). Voice media has also been added to multi-part message bodies.

4.1.2. iFax

This set of profiles ([16], [17], [18], [19], [20], [21]) enables the transport of fax using Internet mail protocols. This work defined the image/tiff MIME type. Support for fax clients also required extensions to Message Delivery Notification.

4.1.3. Internet Voice Mail (IVM) [34]

This proposed mail enhancement (whose requirements are described in RFC 3773 [30]) targets support for the interchange of voice messaging between the diverse components (clients as well as servers) in systems supporting voice mail.

4.2. Putative Client Profiles

4.2.1. TUI

It is desirable to replace proprietary protocols between telephone user interface clients and message stores with standards-based interfaces. The proprietary protocols were created to provide media-aware capabilities as well as to provide the low-latency required by some messaging applications.

An example of a TUI client is a voice mail client. Because a POTS phone lacks any intelligence, the voice mail client functionality has to be provided by a user agent networked to the mail server. The main architectural difference between a conventional voice mail system and an Internet messaging system supporting a TUI is that the voice mail system uses a specialized message store and protocols.

The following figure depicts the architecture of current voice mail systems implementing VPIMv2:

```
                                                  |-------------|
              |-------|     RFC-822/MIME          |             |
              |   |   |---------------------------|     MTA     |
              |   |   |     mail submission ->    |             |(E)SMTP
   Telephone--|TUI|TUA|                           |------|      |-----to
              |   |   |   Proprietary Protocol    |      |      |another
              |   |   |---------------------------| MS   |      | email
              |-------|   < - mail retrieval      |      |      | server
                                                  |-------------|
              mail client                          email server

            |----------------voice messaging system -------------|
```

Mail client consists of: TUI (Telephone User Interface) and
                         TUA (Telephone User Agent)

Communication between TUI and TUA is proprietary.

Email server consists of: MS (Mail Store) and
                          MTA (Message Transfer Agent)

Communication between MS and MTA is proprietary.

It is proposed that the Proprietary Protocol be replaced with an IETF standard protocol:

```
                                                  |-------------|
              |-------|     RFC-822/MIME          |             |
              |   |   |---------------------------|     MTA     |
              |   |   |   mail submission ->      |             |(E)SMTP
   Telephone--|TUI|TUA|                           |------|      |-----to
              |   |   |     IETF protocol         |      |      |another
              |   |   |---------------------------| MS   |      | mail
              |-------|    <- mail retrieval      |      |      | server
                                                  |-------------|
              mail client                          email server

         |- voice mail system-|                   |-mail server-|
```

Mail client consists of: TUI (Telephone User Interface) and
                         TUA (Telephone User Agent)

Communication between TUI and TUA is proprietary.

Email server consists of: MS (Mail Store) and
                          MTA (Message Transfer Agent)

Communication between MS and MTA is proprietary.

4.2.2. Multi-Modal Clients

Multi-modal clients offer the advantage of coordinated voice and data modes of user interaction. Architecturally, the multi-modal client can be considered the union two user agent components -- one a TUI client, the other a simple GUI client. See the next figure. The Graphical User Agent (GUA) helps maintain the text display while the Telephone User Agent (TUA) acts on behalf of the TUI functionality.

This model is the norm with cellular devices supporting data access because historically they evolved from cell phones to which a data channel was added. The presentation of multiple complementary modes of interaction gives end-users their choice of the most convenient and natural working mode for a particular task. There are other situations where a multi-modal model is appropriate. (For example, a telephone sales unit needs to provide a voice (telephone) mode and conventional desktop PC mode of interaction at the same time in an integrated manner.)

A major issue in the design of multi-modal clients -- the need to synchronize the component user agents making up a client -- is only addressed by LEMONADE to a limited extent in Section 6.3.

4.2.3. WUI

The Wireless User Interface is functionally equivalent to a conventional email client on a personal workstation, but is optimized for clients on handheld tetherless devices. Factors needing consideration include limited memory and processing power. Limited bandwidth is also relatively high cost. As already alluded to above, in many cases (e.g., cellular devices), the mobile client is multi-modal. So WUIs can be modeled as resource-and-link-limited multi-modal clients.

These terminals require the use of protocols that minimize the number of over-the-air transactions and reduce the amount of data that need be transmitted over the air overall. Such reduction in over-the-air transmission is a combination of more efficient protocol interaction and richer message presentation choices, whereby a user may more intelligently select what should be downloaded and what should remain on the server.

Although not an explicit goal, providing equivalent or superior functionality to the wireless MMS service [43] (as defined by 3GPP, 3GPP2, and the OMA) is desirable.

```
   Proposed Wireless User Interface (WUI)/Multi-modal Clients

          |wireless GUI client|                   email server

                         (E)SMTP (client-server)  |-------------|
              |-------|     RFC-822/MIME          |             |
              |   |   |---------------------------|             |
              |   |   |   mail submission ->      |             |(E)SMTP
             -|GUI|GUA|                           |             |-----to
            | |   |   | IETF standard protocol    |------------ |another
            | |   |   |----------------------------to MS below| | mail
            | |-------|    <- mail retrieval      |------------ | server
            |       |                             |             |
   Handheld |       |                             |             |
   Device   WUI     |                             |    MTA      |
            |       |                             |             |
            | |-------|     RFC-822/MIME          |             |
            | |   |   |---------------------------|             |
            | |   |   |   mail submission ->      |             |
             -|TUI|TUA|                           |------|      |
              |   |   |  IETF standard protocol   |      |      |
              |   |   |---------------------------| MS   |      |
              |-------|    <- mail retrieval      |      |      |
                                                  |-------------|
              TUI client                          voice mail server

         |----------------voice messaging system ----------------|

         |------WUI-----|                      |---mail server---|

   Wireless GUI client consists of: GUI (Graphical User Interface) and
                                    GUA (Graphical User Agent)

      Communication between UI and UA is proprietary.

   TUI client consists of: TUI (Telephone User Interface) and
                           TUA (Telephone User Agent)

      Communication between TUI and TUA is proprietary.
      Communication between GUA and TUA is proprietary.

   Mail (email and voice mail) server consists of:
                                    MS (Mail Store) and
                                    MTA (Message Transfer Agent)

      Communication between MS and MTA is proprietary.
```

5. General Principles

This is a list of principles to guide the design of extensions for Internet Messaging systems and protocols to support diverse endpoints.

5.1. Protocol Conservation

5.1.1. Reuse Existing Protocols

To the extent feasible, the enhanced messaging framework SHOULD use existing protocols whenever possible.

5.1.2. Maintain Existing Protocol Integrity

In meeting the requirement "Reuse Existing Protocols" (Section 5.1.1), the enhanced messaging framework MUST NOT redefine the semantics of an existing protocol.

Extensions, based on capability declaration by the server, will be used to introduce new functionality where required.

Said differently, we will not break existing protocols.

5.2. Sensible Reception/Sending Context

5.2.1. Reception Context

When the user receives a message, that message SHOULD receive the treatment expected by the sender. For example, if the sender believes he is sending a voice message, voice message semantics should prevail to the extent that the receiving client can support such treatment.

5.2.2. Sending Context

When the user sends a message, he SHOULD be able to specify the message context. That is, whether the network should treat the message as an text message, voice message, video message, etc. Again, this can only be complied with to the extent that the infrastructure and receiving client can provide such treatment. In practice, this would imply that the message should be in the form desired by the sender up to delivery to the receiving client.

5.3. Internet Infrastructure Preservation

The infrastructure SHOULD change only where required for new functionality. Existing functionality MUST be preserved on the existing infrastructure; that is, all extensions must be backward compatible to allow for the gradual introduction of the enhancements. Messages created in an enhanced messaging context MUST NOT require changes to existing mail clients. However, there may be a degradation in functionality in certain circumstances.

The enhanced messaging framework MUST be able to handle messages created in a non-enhanced messaging context; for example, a simple, RFC822 [2] text message.

5.4. Voice Requirements (Near Real-Time Delivery)

On the retrieval side, there are significant real-time requirements for retrieving a message for voice playback. More than any other media type, including video, voice is extremely sensitive to variations in playback latency. The enhanced messaging framework MUST address the real-time needs of voice.

5.5. Fax Requirements (Guaranteed Delivery)

Fax users have a particular expectation that is a challenge for enhanced Internet messaging. A person who sends a fax expects the recipient to receive the fax upon successful transmission. This clearly is not the case for Internet Mail.

Addressing this need is not in the scope of LEMONADE.

5.6. Video Requirements (Scalable Message Size)

Video mail has one outstanding feature: Video messages are potentially large! The enhanced messaging framework MUST scale for very large messages. Streaming from the server to the client, in both directions, MUST be supported.

6. Issues and Requirements: TUI Subset of WUI

6.1. Requirements on the Message Retrieval Protocol

IMAP [10] is the Internet protocol for rich message retrieval and manipulation. The project MUST limit itself to extending IMAP where necessary and MUST not create a new protocol.

6.1.1. Performance Issues

6.1.1.1. Real-Time Playback

The real-time playback of a voice message MUST be supported so that the user experience does not differ noticeably from that of a conventional voice messaging system.

Possible solutions for this include making use of the existing incremental download capability of the IMAP protocol, or utilizing a companion streaming protocol.

The IMAP protocol itself does not provide streaming by the strict definition of the term. It does provide for the incremental download of content in blocks. Most IMAP clients do not support this behavior and instead download the entire contents into a temporary file to be passed to the application.

There are several approaches to achieve real-time playback. The first approach is to implement an IMAP client that can pass data incrementally to the application as it is received from the network. The application can then read bytes from the network as needed to maintain a play buffer. Thus, it would not require the full download of contents. This approach may require server-side development to support partial download efficiently (i.e., to avoid re-opening files and positioning to the requested location).

Alternatively, the client can use the proposed IMAP channel extension [32] to request that the server make the selected content available via an alternate transport mechanism. A client can then ask the server to make the voice data available to the client via a streaming media protocol such as RTSP. This requires support on the client and server of a common streaming protocol.

6.1.1.2. Avoid Content-Transfer-Encoding Data Inflation

Another important performance optimization is enabling the transport of data using more efficient native coding rather than text-like content-transfer encodings such as "base 64".

Standard IMAP4 uses a text-based data representation scheme where all data is represented in a form that looks like text; that is, voice data must be encoded using "base 64" into a transport encoding that adds 30% to the size of a message. Downloading or appending large messages to the server already uses substantial bandwidth.

Possible Solutions:

Where IMAP channel is appropriate, the external channel may be binary capable; that is, the external access may not require re-encoding. Mechanisms such as HTTP [24], FTP, or RTSP are available for this download.

The IMAP binary extension standards proposal [31] extends the IMAP fetch command to retrieve data in the binary form. This is especially useful for large attachments and other binary components. Binary in conjunction with a streaming client implementation may be an attractive alternative to the channel extension.

6.1.2. Functional Issues

6.1.2.1. Mailbox Summary Support

The common TUI prompt, "you have two new voice messages, six unheard messages, and one new fax message", requires more information than is conveniently made available by current message retrieval protocols.

The existing IMAP protocol's mailbox status command does not include a count by message context [26] [27]. A possible solution is for the mail server to keep track of these current counters and provide a status command that returns an arbitrary mailbox summary. The IMAP status command provides a count of new and total messages with standardized attributes extracted from the message headers. This predetermined information does not currently include information about the message type. Without additional conventions to the status command, a client would have to download the header for each message to determine its type, a prohibitive cost where latency or bandwidth constraints exist.

6.1.2.2. Sort by Message Context Support

This functionality is required to present new voice messages first and then new fax messages within a single logical queue as voice mailboxes commonly do. Again, this is a question of convenience and performance. Adequate performance may only be possible if the mail server provides a sort by context or maintains a set of virtual mailboxes (folders) corresponding to message types as for "Mailbox Summary Support", Section 6.1.2.1.

IMAP does not support this directly. A straightforward solution is to define an extensible sort mechanism for sorting on arbitrary header contents.

6.1.2.3. Status of Multiple Mailboxes Support

Extension mailbox support requires the ability to efficiently status a mailbox other than the one currently logged into. This facility is required to support sub-mailboxes, where a common feature is to check whether other sub-mailboxes in the same family group have new messages.

Current mechanisms are limited to logging into each of set of mailboxes, checking status, logging out, and repeating until all sub-mailboxes are processed.

6.1.2.4. Specialized Mailbox Support

Applications that provide features such as check receipt, deleted message recovery, resave, and others, require the ability to access messages in predetermined mailboxes with specific behaviors (e.g., Outbox, Sent Items, Deleted Items, Expired Items, Drafts).

IMAP provides only a single standardized folder, the inbox. This functionality does not require new protocol additions per se, but standardized usage and naming conventions are necessary for interoperability. This functionality requires that the server provide the underlying logic to support these special folders, including automatic insertion, scheduled copying, and periodic deletion.

6.1.2.5. CLID Restriction Indication/Preservation

Many calling features are dependent on collected caller-ID information. Clients -- such as the TUI and other service supporting user agents (e.g., WEB and WAP servers) -- may need trusted access to restricted caller-ID information for such purposes as callback. Untrusted clients must not be permitted to receive this information. A mechanism for establishing "trust" between appropriate clients and the server is required to restrict delivery of this information to the end-user only as allowed.

Further, when messages are sent between servers within a network, a means of communicating trust is needed so that the identity of the sender can be preserved for record-keeping and certain features while ensuring that the identity is not disclosed to the recipient in an inappropriate way.

6.1.2.6. Support for Multiple Access to Mailbox

If the telephone answering application client uses IMAP4 for greeting access and message deposit, it is essential that the server provide support for simultaneous login. It is common in voice mail for an incoming call to be serviced by the telephone answering application client at the same time the subscriber is logged into her mailbox. Further, new applications such as WEB and WAP access to voice mail may entail simultaneous login sessions, one from the TUI client and one from the visual client.

The existing standard does not preclude multiple accesses to a mailbox, but it does not explicitly require support of the practice. The lack of explicit support requires the server and client to adhere to a common set of practices and behaviors to avoid undesirable and unpredictable behaviors. RFC2180 [29] describes a candidate set of conventions necessary to support this multiple-access technique. It or some other method MUST be standardized as part of LEMONADE.

6.2. Requirements on the Message Submission Protocol [22]

6.2.1. Forward without Download Support

It is common to forward messages or to reply to messages with a copy of their attached content. Today such forwarding requires the sender to download a complete copy of the original message, attach it to the reply or forward message, and resubmit the result. For large messages, this represents a substantial amount of bandwidth and processing. For clients connected via long-thin pipes, alternatives are required.

One approach is to define an extension to message submission to request the submission server to resolve embedded URLs within a message before relaying the message to the final destination. This approach is referred to as the pull approach because the message submission server must pull data from the IMAP server.

Another approach is to add a limited message assembly and submission capability to the IMAP server. This approach muddies the distinction between the message submission protocol and that for message storage and retrieval (IMAP) because now message submission may be a side effect of message store commands. This approach is referred to as the push approach because in this case the IMAP server pushes data to the message submission server.

A detailed analysis of which of the two approaches is preferable as well as implementation details of both can be found in references [36], [37], [38], [39], [40], and [41].

6.2.2. Quota by Context Enforcement

It is common in a unified messaging system to offer separate quotas [11] for each of several message contexts to avoid the condition where a flood of email fills the mailbox and prevents the subscriber from receiving voice messages via the telephone. It is necessary to extend the protocols to support the reporting of the "mailbox full" status based on the context of the submitted message.

An obvious security issue needing consideration is the prevention of the deliberate misidentification of a message context with the intention of overflowing a subscriber's mailbox. It is envisioned that the message submission protocol will require the authentication of trusted submission agents allowing only those so authorized to submit distinguished messages.

Voice mail system mailboxes commonly contain voice and fax messages. Sometimes, such systems also support email messages (text, text with attachments, and multimedia messages) in addition to voice messages. Similar to the required sort by message context, quota management is also required per message context.

One possible use case is the prevention of multiple (large) messages of one type (e.g., email messages) from consuming all available quota. Consumption of all quota by one type prevents the delivery of other types (e.g., voice or fax messages) to the mailbox.

One possible approach is to define a mechanism whereby a trusted client can declare the context of a message for the purpose of utilizing a protected quota. This may be by extensions to the SMTP-submit or LMTP[35] protocols.

6.2.3. Future Delivery Support with Cancel

Traditionally messages sent with "future delivery" are held in the recipient's client "outbox" or its equivalent until the appointed submission time. Thin clients used with TUIs do not have such persistent storage or may be intermittently connected and must rely upon server-based outbox queues.

Such support requires extensions to message submission protocols to identify a message as requiring queuing for future delivery. Extensions to IMAP4 or SMTP are required for viewing and manipulating the outbound queue, for such purposes as canceling a future message. Server support for managing such a queue is required so that messages are sent when they are intended.

Some of the architectural issues here are the same as those in "Forward without Download Support" (Section 6.2.1).

6.2.4. Support for Committed Message Delivery

Voice messaging service has provided a high degree of reliability and performance for telephone answering messages. The expectation is that once the caller has hung up, the message is in the mailbox and available for review. The traditional Internet mail architecture suggests these messages should be sent to the mailbox via SMTP. This approach has two limitations. The first and most manageable is that the message forwarding may take more time than is tolerable by the subscriber. The second is that the message may fail to be delivered to the mailbox. Because there is no way to return notice to the caller, the message is "lost".

The standards community is working on an alternative to SMTP called Local Message Transport Protocol (LMTP[35]). This protocol addresses a number of limitations in SMTP when used to provide atomic delivery to a mailbox. The failure modes in this proposal are carefully controlled, as are issues of per-message quota enforcement and message storage quota-override for designated administrative messages.

An alternative approach is to misuse the IMAP protocol and use an IMAP-based submission mechanism to deposit a message directly into the recipient's inbox. This append must be done by a special super-user with write permissions into the recipient mailbox. Further, the message store must be able to trigger notification events upon insertion of a message into the mailbox via the Append command. The historic limitation on using IMAP4 for message sending involves the inability of IMAP to communicate a full SMTP envelope. For telephone answering, these limitations are not significant. However, the architectural issues raised by this approach are significant. See "Forward without Download Support" (Section 6.2.1).

6.3. Requirements on Message Notification

Clients keep local information about the IMAP store. This information must be kept synchronized with the state of the store.

For example, voice mail systems traditionally notify subscribers of certain events happening in their mailbox. It is common to send an SMS or a pager notification for each message arrival event, message read event, mailbox full event, etc.

When implemented over IMAP-based message stores, the voice mail client needs to be notified about these events. Furthermore, when other applications access/manipulate the store, these events need to be communicated to the mail client. In some cases, the client needs to notify the user immediately. In most cases, it is a question of maintaining client/application consistency. In the case of a multimodal client, it is especially important to provide a means of coordinating the client's different modal views of the state of the store.

Email systems have traditionally polled to update this information. There may be advantages to an event-driven approach in some cases.

The standards community is working on a standard for bulk server-to-client status notification. An example of such work is the Simple Notification and Alarm Protocol (SNAP) [45], which defines the expected behavior of the message store for various events, many of them triggered by IMAP commands.

6.3.1. Additional Requirements on Message Notification

A format for message notification for servers reporting status information to other servers (e.g., IMAP4 server to SMS or pager server) MUST be defined. The method for delivery of these notifications MUST also be specified.

The design for this MUST take into account the IAB note: "Unified Notification Protocol Considerations" (Appendix C).

7. Issues and Requirements: WUI Mobility Aspects

7.1. Wireless Considerations on Email

7.1.1. Transport Considerations

Compared to a LAN/WAN configuration or even to a wire-line dial-up connection, the probability of an interruption to a wireless connection is very high.

Interruptions can be due to handoff, signal fading, or stepping beyond cell coverage.

In addition, because the mobile handset is also used for other types of communications, there is a relatively high probability that the data session will be interrupted either by incoming voice calls or by "pushed" messages from services such as SMS, MMS, and WAP.

It is also common in these environments that the device's IP address change within a session.

7.1.2. Handset-Resident Client Limitations

Although the capabilities of wireless handsets are rapidly improving, the wireless handset remains limited in its capability to host email clients. Currently, email access is restricted to only high-end wireless handsets.

These limitations include:

- o Client size
  Handset-resident clients are limited in size because either the handset has limited storage space or the handset vendor/network operator has set a limit on the size of client application that can reside on the handset.
- o Runtime memory
  Wireless handsets have limited runtime memory for the use of the mobile email client.
- o CPU Speed
  Wireless handsets have CPUs that are inferior to those in conventional systems (PCs) that run email clients.
- o User Interface
  Handsets have very limited input and output capabilities. Most of them have only a rudimentary keyboard (a keypad) and a rudimentary pointing device (a text cursor).

7.1.3. Wireless Bandwidth and Network Utilization Considerations

7.1.3.1. Low Bandwidth

2G mobile networks enabled wireless data communications, but only at very low bandwidths using circuit-switched data. 2.5G and 3G networks improve on this. However, existing email clients require very large files (up to several MBs) -- encountered in multi-media attachments such as presentations, images, voice, and video -- to be downloaded even though mobiles cannot exploit most of the data (because of color depth and screen size limitations). Transferring such large files over the air is of questionable value even when higher wireless bandwidth is available.

7.1.3.2. Price Sensitivity

In many cases, users of mobile data services are charged by the amount of data (e.g., kilobytes) downloaded to the handset. Most users currently experience a higher per-kilobyte data charge with a wireless service than they do over a wire-line service. Users are sensitive to the premium for wireless service. This results in an unwillingness to download large amounts of unnecessary data to the handset and the desire to be able to download only selected content.

7.1.3.3. File Size Limitations

In some cases, the size of file that can be transmitted over the air to the handset is limited. This is a consequence of handset limitations (Section 7.1.2), wireless media and bandwidth issues (Section 7.1.1 and Section 7.1.3.1), and price sensitivity (Section 7.1.3.2).

7.1.4. Content Display Considerations

7.1.4.1. Display Size and Capabilities

Wireless terminals are currently limited in their display size, color depth, and ability to present multimedia elements (i.e., if multiple pictures are sent, the mobile can usually present only one reduced-sized picture element at a time rather than the several picture elements at once in the same display that a conventional PC email client would be able to show). Therefore, many email attachments destined for a mobile may require changes in size, color depth, and presentation method in order to be suitably displayed.

7.1.4.2. Supported Media Formats

Wireless handsets can only display a limited set of media format types. Although PC clients support a large variety of document types (and allow on-demand "codec"/player download), mobiles have very limited support. (For example, most only support WAV audio and cannot play other formats such as AU, MP3 and AIFF.) Furthermore, although almost all new handsets sold today can display images and sound in some advanced format, support for displaying other media or application-specific formats, such as MS Office (TM), is not expected to be widespread in the near future.

7.1.4.3. Handset Type Variety

As mentioned above, there are many handset types available in the market, and each has different display capabilities, screen characteristics, and processing capabilities. The mobile email service should be able to support as many handset types as possible.

7.1.4.4. Specific Attachment Display Scenarios

Handsets are unsuitable for perusing entire lengthy documents or presentations. Rather than go through the whole document, a mobile user is more likely to look at several pages of a document or several slides of a presentation and then take action accordingly (e.g., forward the email message to another recipient, print it, or leave the document for later retrieval from another device).

Therefore, there is a need to enable users to download not the entire attachment but rather just a selected part of it. For example, users should be able to download the "Table of Contents" of a document; to search within a document; to download the first slide of a presentation; the next slide of this presentation or a range of slides, etc.

7.2. Requirements to Enable Wireless Device Support

The following requirements are derived from the considerations mentioned above.

7.2.1. Transport Requirements

The mobile email protocol must anticipate transient losses of connectivity and allow clients to recover (restore state) from interrupted connections quickly and easily.

IMAP4 Context

An IMAP4 connection requires the communication socket to remain up continuously during an email session. In case of transient loss of communications, the connection must be reestablished. It is up to the client to reconnect to the server and return to an equivalent state in the session. This overhead of restoring connections is very costly in response time and additional data transmission.

7.2.2. Enhanced Mobile Email Functionality

7.2.2.1. Forward without Fetch

To minimize the downloading of data over the air, the user MUST be able to forward a message without initially downloading it entirely or at all to the handset.

The mobile email protocol MUST support the ability to forward a message without retrieving it.

This requirement is identical to the TUI requirement described in "Forward Without Download Support" (Section 6.2.1).

7.2.2.2. Media Streaming

The mobile email protocol MUST provide a solution that will enable media streaming to the wireless handset.

This requirement is similar to the TUI requirement described in "Real-Time Playback" (Section 6.1.1.1).

7.2.3. Client Requirements

IMAP4 clients are large because IMAP4 already consists of a complex set of functions (e.g., parsing of a broad variety of MIME formats).

The mobile email client should be:

- o Small in size
- o Efficient in CPU consumption
- o Efficient in runtime memory consumption

To enable such extremely thin clients, in developing the mobile email protocol we should consider simplifying the IMAP functionality that handsets need to support. However, any such simplification MUST NOT limit interoperability with full IMAP servers.

7.2.4. Bandwidth Requirements

The mobile email solution should minimize the amount of data transmitted over the air. There are several ways of pursuing this goal that can be used in conjunction.

One way is the use of content transcoding and media adaptation by the server before message retrieval in order to optimize the message for the capabilities of the receiving handset.

Another possible optimization is to make the mobile email protocol itself simple, containing as little overhead as possible.

A third approach is to minimize the bandwidth usage as described in "Avoid Content-Transfer-Encoding Data Inflation" (Section 6.1.1.2).

7.2.5. Media Handling Requirements

As described above, wireless devices have limited ability to handle media. Therefore, the server may be have to perform media manipulation activities to enable the terminal to display the data usefully.

7.2.5.1. Device Capabilities Negotiation

In order to support the different characteristics and capabilities of the various handset types available in the market correctly, the mobile email protocol must include provision for email content adaptation. For example, the choice of supported file formats, color depth, and screen size. Work on ESMTP transcoding (CONNEG[33]) may address this issue.

7.2.5.2. Adjusting Message Attachments for Handset Abilities

To support wireless handsets, the server could transcode the message attachments into a representation that is more suitable for that device. This behavior should be based on the device capabilities negotiation as described in "Device Capabilities Negotiation" (Section 7.2.5.1). For example, a device that cannot display GIF format, and can only display WBMP, should get a WBMP image. Devices that cannot display a PDF file should get a text version of the file.

The handset should control what transcoding, if any, is desired. It should be able to retrieve the original attachment without any changes. In addition, the device should be able to choose between "flavors" of the transcoding. ("Present the content as thumbnail image" is an example of such a specific media manipulation.)

Again, work on ESMTP transcoding (CONNEG[33]) may address this issue.

7.2.5.3. Handling Attachment Parts

A desirable feature (but out of scope for the current LEMONADE charter) is to enable users the choice of retrieving parts of an attachment file, not just the entire attachment. The mobile email protocol should include the ability for the retrieving client to specify selected elements of an attachment for download. Such elements can be, for example, specific pages of a document, the "table of contents" of a document, or specific slides of a presentation.

8. Interoperation with Existing Mobile Messaging

LEMONADE's charter includes the specification of how enhanced Internet mail will interoperate with existing mobile messaging services (e.g., MMS) to deliver messages to mobile clients.

8.1. Addressing of Mobile Devices

E.164 addressing [62] is prevalent in mobile messaging services to address recipient mobiles. Consideration should be given to supporting E.164 addressing for mobile devices in addition to RFC822 addressing.

8.2. Push Model of Message Retrieval [49] [50] [51]

MMS provides a "push" option for message retrieval. The option hides network latencies and reduces the need for user-handheld interaction. If a level of support for mobiles comparable to that of MMS is desired, this mode of operation should be considered.

8.3. Message Notification [44] [55]

Message notification was alluded to in "Requirements on Message Notification" (Section 6.3). Internet mail has not so far standardized a server-to-client notification protocol although most existing wireless mail systems use notification to avoid needless polling. Client-to-server notification is not within the LEMONADE charter.

8.4. Operator Issues

8.4.1. Support for End-to-End Delivery Reports and Message-Read Reports

Support for committed delivery is described in Section 6.2.4, but this is different.

8.4.2. Support for Selective Downloading

If a push model of message retrieval is supported, the need for selective downloading and SPAM control is especially important.

8.4.3. Transactions and Operator Charging Units

Mobile network providers often operate on a "pay for use" service model. This brings in requirements for clearly delineated service transactions that can be reported to billing systems, and for positive end-to-end acknowledgement of delivery or non-delivery of messages already mentioned in Section 8.4.1. Note that billing is specifically outside the scope of the IETF.

8.4.4. Network Authentication

Some mobile networks require network authentication as well as application authentication.

8.5. LEMONADE and MMS

The 3GPP MMS Reference Architecture ([48] [54]) defines seven interfaces labelled MM1 to MM7, as below:

```
                   3GPP MMS Reference Architecture (subset)

            |---------|                          |------------|
   wireless ||-------||                          |            |
    device  || MMS   ||                          |            |<- MM2 ->
            || USER  |---------------------------|            |---------
            || AGENT |<-         MM1           ->|            | to
            ||-------||                          |            | another
            |---------|                          |            | MMS
                                                 |            | relay/
             |--------|                          |            | server
      e.g.,  |        |                          |            |
      Email, |EXTERNAL|                          |            |
      Fax, or| SERVER |--------------------------|            |
      UMS    |        |<-        MM3           ->|            |
             |--------|                          |            |
                                                 |            |
             |---------|                         |            |
             |"FOREIGN"|                         |            |
             | MMS     |-------------------------|            |
             | relay/  |<-       MM4           ->|            |
             | server  |                         |            |
             |---------|                         |            |
                                                 |    MMS     |
             |-------|                           |relay/server|
             |       |                           |            |
             |  HLR  |---------------------------|            |
             |       |<-         MM5           ->|            |
             |-------|                           |            |
                                                 |            |
             |-------|                           |            |
             |  MMS  |                           |            |
             |  USER |---------------------------|            |
             |  DBs  |<-         MM6           ->|            |
             |-------|                           |            |
                                                 |            |
             |-------|                           |            |
             |  MMS  |                           |            |
             |  VAS  |---------------------------|            |
             |  APPs |<-         MM7           ->|            |
             |-------|                           |------------|

       MMS - Multimedia Messaging Service
       UMS - Unified Messaging Service
       HLR - Home Location Register
       DB  - Data Base
       VAS - Value Added Service
       APP - Application
```

The LEMONADE profile provides an enhanced IMAP mail retrieval protocol suitable for use at interfaces MM1 and MM3.

In addition, if the wireless device uses a LEMONADE-enhanced IMAP user agent, the enhanced IMAP protocol can be used to access Internet mail directly, as below.

                   3GPP MMS Reference Architecture (subset)

```
            |---------|                          |------------|
   wireless ||-------||                          |            |
    device  || IMAP  ||                          |            |<- MM2 ->
            || USER  ||                          |            |---------
            || AGENT ||                          |            | to
            ||---^---||                          |            | another
            |----|---||                          |            | MMS
                 | LEMONADE Enhanced IMAP and    |            | relay/
             |---V----|          SMTP            |            | server
      e.g.,  |        |                          |            |
      Email, |EXTERNAL|                          |            |
      Fax, or| SERVER |--------------------------|            |
      UMS    |        |<-        MM3           ->|            |
             |--------|                          |            |
                                                 |            |
             |---------|                         |            |
             |"FOREIGN"|                         |            |
             | MMS     |-------------------------|            |
             | relay/  |<-       MM4           ->|            |
             | server  |                         |            |
             |---------|                         |            |
                                                 |    MMS     |
             |-------|                           |relay/server|
             |       |                           |            |
             |  HLR  |---------------------------|            |
             |       |<-         MM5           ->|            |
             |-------|                           |            |
                                                 |            |
             |-------|                           |            |
             |  MMS  |                           |            |
             |  USER |---------------------------|            |
             |  DBs  |<-         MM6           ->|            |
             |-------|                           |            |
                                                 |            |
             |-------|                           |            |
             |  MMS  |                           |            |
             |  VAS  |---------------------------|            |
             |  APPs |<-         MM7           ->|            |
             |-------|                           |------------|
```

MMS	-	Multimedia Messaging Service
UMS	-	Unified Messaging Service
HLR	-	Home Location Register
DB	-	Data Base
VAS	-	Value Added Service
APP	-	Application

9. Security Considerations

Security will be a very important part of enhanced messaging. The goal, wherever possible, is to preserve the semantics of existing messaging systems and to meet the (existing) expectations of users with respect to security and reliability.

10. References

10.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. Informative References

[2] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[3] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[4] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[5] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[6] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[7] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text ", RFC 2047, November 1996.

[8] Freed, N., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 2048, November 1996.

[9] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.

[10] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[11] Myers, J., "IMAP4 QUOTA extension", RFC 2087, January 1997.

[12] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.

[13] Vaudreuil, G. and G. Parsons, "Voice Profile for Internet Mail - version 2 (VPIMv2)", RFC 3801, June 2004.

[14] Vaudreuil, G. and G. Parsons, "Toll Quality Voice - 32 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM) MIME Sub-type Registration", RFC 3802, June 2004.

[15] Vaudreuil, G. and G. Parsons, "Content Duration MIME Header Definition", RFC 3803, June 2004.

[16] Buckley, R., Venable, D., McIntyre, L., Parsons, G., and J. Rafferty, "File Format for Internet Fax", RFC 3949, February 2005.

[17] Parsons, G. and J. Rafferty, "Tag Image File Format (TIFF) - image/tiff MIME Sub-type Registration", RFC 3302, September 2002.

[18] Allocchio, C., "Minimal GSTN address format in Internet Mail", RFC 3191, October 2001.

[19] Allocchio, C., "Minimal FAX address format in Internet Mail", RFC 3192, October 2001.

[20] Toyoda, K., Ohno, H., Murai, J., and D. Wing, "A Simple Mode of Facsimile Using Internet Mail", RFC 3965, December 2004.

[21] Parsons, G. and J. Rafferty, "Tag Image File Format (TIFF) - F Profile for Facsimile", RFC 2306, March 1998.

[22] Gellens, R. and J. Klensin, "Message Submission", RFC 2476, December 1998.

[23] Masinter, L. and D. Wing, " Extended Facsimile Using Internet Mail", RFC 2532, March 1999.

[24] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[25] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[26] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[27] Burger, E., Candell, E., Eliot, C., and G. Klyne, "Message Context for Internet Mail", RFC 3458, January 2003.

[28] Burger, E., "Critical Content Multi-purpose Internet Mail Extensions (MIME) Parameter", RFC 3459, January 2003.

[29] Gahrns, M., "IMAP4 Multi-Accessed Mailbox Practice", RFC 2180, July 1997.

[30] Candell, E., "High-Level Requirements for Internet Voice Mail", RFC 3773, June 2004.

[31] Nerenberg, L., "IMAP4 Binary Content Extension", RFC 3516, April 2003.

[32] Nerenberg, "IMAP4 Channel Transport Mechanism", Work in Progress, November 2001.

[33] Toyoda, K. and D. Crocker, "SMTP Service Extensions for Fax Content Negotiation", Work in Progress, February 2003.

[34] McRae, S. and G. Parsons, "Internet Voice Messaging (IVM)", RFC 4239, November 2005.

[35] Murchison, K. and L. Greenfield, "LMTP Service Extension for Ignoring Recipient Quotas", Work in Progress, June 2002.

[36] Crispin, M., "Message Submission", Work in Progress, February 2004.

[37] Newman, C., "Message Submission with Composition", Work in Progress, February 2004.

[38] Gellens, R., "IMAP Message Submission", Work in Progress, December 2003.

[39] Resnick, P., "Internet Message Access Protocol (IMAP) CATENATE Extension", Work in Progress, December 2003.

[40] Crispin, M. and C. Newman, "Internet Message Access (IMAP) - URLAUTH Extension", Work in Progress, July 2004.

[41] Newman, D., "Message Submission BURL Extension", Work in Progress, July 2004.

[42] Crocker, D., "Internet Mail Architecture", Work in Progress, July 2004.

[43] Leuca, I., "Multimedia Messaging Service", Presentation to the VPIM WG, IETF53 Proceedings , April 2002.

[44] Mahy, R., "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", RFC 3842, August 2004.

[45] Shapira, N. and E. Aloni, "Simple Notification and Alarm Protocol (SNAP)", Work in Progress, December 2001.

[46] Vaudreuil, G., "Messaging profile for telephone-based Messaging clients", Work in Progress, February 2002.

[47] Burger, E., "Internet Unified Messaging Requirements", Work in Progress, February 2002.

[48] OMA, "Multimedia Messaging Service Architecture Overview Version 1.1", Open Mobile Alliance (OMA) OMA-WAP-MMS-ARCH-v1_1-20021101-C, November 2002.

[49] OMA, "Push Architectural Overview", Open Mobile Alliance (OMA) WAP-250-PushArchOverview-20010703-a, July 2001.

[50] OMA, "Push Access Protocol Specification", Open Mobile Alliance (OMA) WAP-247-PAP-20010429-a, April 2001.

[51] OMA, "Push Proxy Gateway Service Specification", Open Mobile Alliance (OMA) WAP-249-PPGService-20010713a, July 2001.

[52] OMA, "Multimedia Messaging Service; Client Transactions Version 1.1", Open Mobile Alliance (OMA) OMA-WAP-MMS-CTR-v1_1-20021031-C, October 2002.

[53] OMA, "Multimedia Messaging Service; Encapsulation Protocol Version 1.1", Open Mobile Alliance (OMA) OMA-MMS-ENC-v1_1-20021030-C, October 2002.

[54] OMA, "User Agent Profile, Version 1.1", Open Mobile Alliance (OMA) OMA-UAProf-v1_1-20021212-C, December 2002.

[55] OMA, "Email Notification Version 1.0", Open Mobile Alliance (OMA) OMA-EMN-v1_0-20021031-C, October 2002.

[56] 3GPP, "Third Generation Partnership Project; Technical Specification Group Services and System Aspects; Service aspects; Functional description; Stage 1 Multimedia Messaging Service", 3GPP TS 22.140, 2001.

[57] 3GPP, "Third Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2", 3GPP TS 23.140, 2001.

[58] 3GPP2, "Short Message Service (SMS)", 3GPP2 TSG C.S0015-0, December 1999.

[59] 3GPP2, "Enhanced Message Service (EMS) Stage 1 Description", 3GPP2 TSG S.R0051-0 v1.0, July 2001.

[60] CCITT, "Recommendations Q.700-Q.716: Specifications of Signalling System No. 7", CCITT White Book, Volume VI, Fascicle VI.7.

[61] CCITT, "Recommendations Q.721-Q.766: Specifications of Signalling System No.7", CCITT White Book, Volume VI, Fascicle VI.8.

[62] ITU, "E.164: The international public telecommunication numbering plan", ITU-T Recommendations Series E, May 1997.

[63] ITU, "Specifications of Signalling System Number 7", ITU White Book, ITU-T Recommendation Q.763.

[64] ITU, "Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit", ITU-T Recommendation X.25, October 1996.

[65] BELLCORE, "Specifications of Signalling System Number 7", GR-246-CORE Issue 1, December 1994.

Appendix A. Contributors

Eric Burger
Brooktrout Technology, Inc.
18 Keewaydin Dr.
Salem, MA 03079
USA

Phone: +1 603 890-7587
EMail: eburger@brooktrout.com

Yair Grosu
Comverse
29 Habarzel St.
Tel-Aviv 69710
Israel

EMail: Yair.Grosu@comverse.com

Glenn Parsons
Nortel Networks
P.O. Box 3511 Station C
Ottawa, ON K1Y 4H7
Canada

Phone: +1 613 763-7582
EMail: gparsons@nortelnetworks.com

Milt Roselinsky
Openwave Systems, Inc.
530 E. Montecito St.
Santa Barbara, CA 93103
USA

Phone: +1 805 884-6207
EMail: milt.roselinsky@openwave.com

Dan Shoshani
Comverse
29 Habarzel St.
Tel-Aviv 69710
Israel

EMail: Dan.Shoshani@comverse.com

Alan K. Stebbens
Openwave Systems, Inc.
530 E. Montecito St.
Santa Barbara, CA 93103
USA

Phone: +1 805 884-3162
EMail: alan.stebbens@openwave.com

Gregory M. Vaudreuil
Lucent Technologies
7291 Williamson Rd.
Dallas, TX 75214
USA

Phone: +1 214 823-9325
EMail: GregV@ieee.org

Appendix B. Acknowledgements

Ari Erev and Noam Shapira (both from Comverse) contributed substantial requirements for IMAP to support a telephone-based (TUI) messaging client. Meir Mendelovich (Comverse) helped in merging the wireless requirements section. Benjamin Ellsworth (Openwave) contributed to mobile messaging architectures and requirements. Yaacov (Jerry) Weingarten (Comverse) and Stephane Maes (Oracle) provided detailed comments on the final document.

Appendix C. IAB Note: Unified Notification Protocol Considerations

Note: dated July 10, 2003

This note was formulated in response to an informal IESG request to look at the architectural issues surrounding a unified notification protocol. The following materials were used as reference:

* draft-dusseault-s2s-event-reqs-00.txt (notification requirements)
* meeting notes for the LEMONADE WG from IETF 56.
* draft-shapira-snap-05.txt (protocol design for SNAP which has some aspects of a generic notification protocol)
* the LEMONADE WG charter
* Recent email on the Lemonade list
* A few presentations from the 1998 UCI workshop on Internet-wide notification

* The Web pages for KnowHow, a company founded by Rohit Khare which has a proprietary Internet-wide notification system.

Thanks to Lisa Dusseault for providing these references.

Note that this opinion does not represent IAB concensus, it is just the opinion of the author after having reviewed the references.

After the reviewing the material, it seemed that the same kinds of functionality are being asked from a generic notification protocol as are asked of desktop application integration mechanisms, like OLAY/COM on Windows or like Tooltalk was on Solaris, but at the level of messaging across the Internet. The desire is that various distributed applications with different application specific mechanisms should be able to interoperate without having an n x n problem of having each application interact with each other application. The cannonical example, which is in a presentation by Lisa Dusseault to LEMONADE from IETF 56, is sending a notification from one application, like XMPP Instant Messaging, and having it delivered on whatever device the recipient happened to be using at the time, like SMS on a cell phone.

The usual problem with application intergration mechanisms on the desktop is how to get the various applications to actually use the mechanism. For Windows, this is relatively easy, since most application developers see major value-added in their applications being able to play nicely with Microsoft Office. For Tooltalk, unfortunatly, Solaris developers didn't see the 10x improvement, and so it was not used outside of Sun's internally maintained applications and a few flagship applications like Framemaker. If the generic notification mechanism requires application developers and other notification protocol designers to make a major effort to utilize it, including modifying their applications or protocols in some way, the protocol could become "just another notification mechanism" rather than a unifying device, because most application developers and other protocol designers could ignore it.

So the first architectural consideration is how do clients of a particular protocol (and the word "client" is used here to mean "any entity using the protocol", they may peers or they may be client/server) actually utilize the generic notification protocol? Is there some code change required in the client or can a legacy client interoperate without change?

If you look at Fig. 1 in draft-shapira-snap-05.txt, the answer seems to be that the notifying client uses the generic protocol, SNAP in this case, to a functional entity (server? module on the receiving client?) called the "Notification Service" that processes the generic notification into an application specific notification and sends that notification to the client. From this figure it looks as if the notifying client would require modification but the receiving client wouldn't.

Another characteristic of application integration mechansims is that they typically focus on very simple operations, the semantics of which are shared between different applications. Examples are "here's a rectangle, display yourself in it" or "put this styled text object into the clipboard", and applications agree on what styled text means. More complicated semantics are hard to share because each application has its own particular twist on the meaning of a particular sequence of operations on a collection of objects. The result is a "least common denominator" collection of integration mechanisms, primarily focussed on display integration and, to a lesser extent, cut and paste integration.

In the context of a generic notification protocol, this raises several possible issues. One is addressing, which is identified draft-dusseault-s2s-event-reqs-00.txt, but in a sense this is the easiest to resolve, by using existing and perhaps newly defined URIs. A more complex problem is matching the semantics of what preconditions constitute the trigger for an event across different application notification mechanisms. This is of course necessary for translating notifications between the different event notification mechanisms and the generic mechanism, but, more problematically, it is also required for a subscription service whereby subscriptions can be made to filter events using the generic notification mechanism and the subscriptions can be translated to different application specific mechanisms. Any language for expressing generic subscriptions is unlikely to support expressing the fine points in the different application notification semantics. Note that SNAP does not seem to support a subscription service so perhaps this isn't an issue for SNAP.

Another architectural issue, which was discussed earlier this year on the LEMONADE list w.r.t. some other topics, is gatewaying. The cannonical example above (message sent using XMPP and arriving via SMS on a cell phone) is actually a gateway example, because it would require translation between an IP-based messaging mechanism (XMPP) to a PSTN based mechanism (SMS). The problem with using a unified notification mechanism for this purpose is that if there are other functions common between the two, it is likely that a gateway will be built anyway. In fact, one of the work items for LEMONADE is to investigate such gateways. The value of a generic notification mechanism therefore needs to be assessed in the light of this.

These are the primary architectural issues, but there are a few others that need consideration in any major system development effort. End to end security is one, draft-dusseault-s2s-event-reqs-00.txt talks about this quite extensively, so it won't be repeated here. The major issue is how to ensure that the end to end security properties are maintained in the face of movement of the notification through the generic intermediary protocol. Another issue is scalability. Peer to peer v.s. server based mechanisms have implications for how scalable the notification mechanism would be, and this needs consideration. Extensibility needs careful consideration. What is required to integrate a new application? Ideally, with time, application developers will stop "rolling their own" notification service and simply use the generic service, but this ideal may be extremely hard to achieve, and may depend to a large extent on market acceptance.

Finally, there are some considerations that aren't architectural but may impact the ultimate success of a generic notification protocol, in the sense that the protocol becomes widely deployed and used. The author's experience is that IETF has not had particular success in introducing mechanisms that unify or supplant existing proprietary mechanisms unless strong vendor and service provider by-in is there. Two examples are instant messaging and service discovery. With instant messaging, it seems that a standarized, unified instant messaging protocol has been delayed by the lack of committment from major service providers. With service discovery, weak commitment from vendors has resulted in the continued introduction of vendor specific service discovery solutions even after an IETF standard is in place. The situation with service discovery (with which the author is most familiar) resulted from a lack of major vendor committment during the end phases of the standarization process. Applying these lessions to a generic notification protocol, having important players with proprietary notification protocols on board and committed until the conclusion of the design process will be crucial. Major committment is needed from various application notification protocols before a generic mechanism could succeed. Given the amount of time and effort required in any IETF standardization work, assessing these with an objective eye is critical, otherwise, regardless of how technically well designed the protocol is, deployment success may be lacking. Having an elegently design solution that nobody deploys is an outcome that might be wise to avoid.

James Kempf
July 2003

Author's Address

Jin Kue Wong (Editor)
Nortel Networks
P.O. Box 3511 Station C
Ottawa, ON K1Y 4H7
Canada

Phone: +1 613 763-2515
EMail: j.k.wong@sympatico.ca

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).