Lemonade
Internet-Draft
Expires: November 20, 2006

N. Cook
Openwave
May 19, 2006

Streaming Multimedia Messaging Attachments
draft-cook-lemonade-streaming-00

Status of this Memo

Copyright Notice

Abstract

This document, part of the Lemonade Working Group, describes the Best Current Practice for streaming multimedia attachments received by a resource constrained and/or mobile device from a IMAP server. It allows such clients, which are limited in storage space and bandwidth, to play video and audio content which is attached to email messages. The document describes a profile for making use of the existing IMAP URLAUTH extension [2] and the "Netann" SIP Media Service as described in RFC 4240 [1].

Conventions Used in this Document

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in RFC 2119 [7].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

1. Introduction

Email clients on resource and/or network constrained devices, such a mobile phones, may have difficulties in retrieving and/or storing large attachments received in a message. For example, on a poor network link, the latency required to download the entire attachment may not be acceptable to the user. Conversely, even on a high-speed network, the device may not have enough storage space to secure the attachment once retrieved.

For certain media, such as audio and video, there is a solution: the media can be streamed to the device, using protocols such as SIP [5] and particularly the media server profile as specified in RFC 4240 [1]. Streaming the media to the device addresses both the latency issue, since the client can start playing the media immediately, and the storage issue, since the client does not need to store the media locally. A tradeoff is that the media cannot be viewed/played when the device is offline.

Examples of the types of media that would benefit from the ability to stream such media to the device include:

+ Voice or Video mail messages received as an attachment

+ Audio clips such as ringtones received as an attachment

+ Video clips such as movie trailers received as an attachment

2. Mechanism

2.1. Overview of Mechanism

The proposed mechanism for streaming media to messaging clients is a profile for making use of two existing mechanisms, namely:

1. IMAP URLAUTH Extension [2] - Providing the ability to generate an IMAP URL [4] which allows anonymous access from external systems to specific message parts; for example, an audio clip.

2. Media Server Announcement Service RFC 4240 [1] - Providing the ability for a media server to stream media using a reference provided by the media server client in a URL.

However, it should be noted that this document proposes a extension to the SIP Parameter Registry [8], in order to accomodate the passing of a content transfer encoding parameter.

The approach is shown in the following figure:

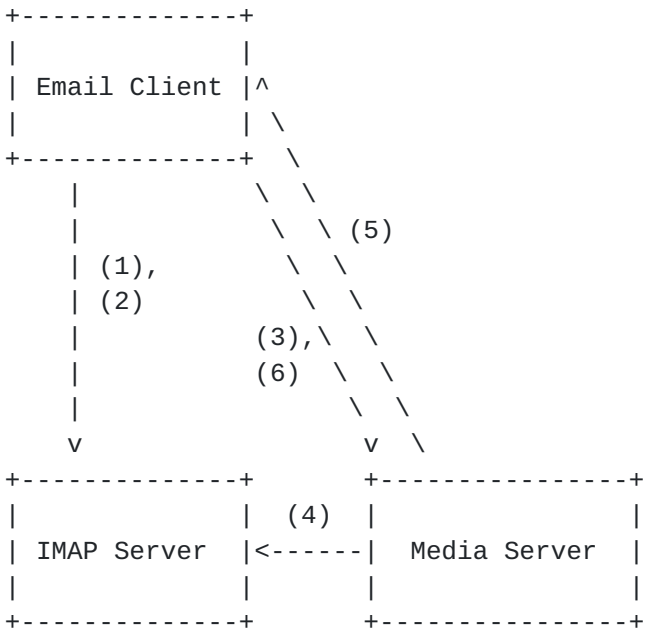


Figure 1

The proposed mechanism has the following steps:

1. Client determines from headers of a particular message that a particular message part (attachment) should be streamed to the

user. Note that no assumptions are made about how/when/if the client contacts the user of the client about this decision. User input MAY be required in order to initiate the proposed mechanism.

2. Client constructs an IMAP URL referencing the message part, and uses the GENURLAUTH [2] command to generate a URLAUTH authorized IMAP URL.

3. Client connects to a SIP Media Server using the Announcement Service as specified in RFC 4240 [1], and passes the URLAUTH authorized URL to the media server.

4. Media Server connects to the IMAP Server specified in the referenced URL, and uses the IMAP URLFETCH [2] command to retrieve the message part.

5. Media server streams the retrieved message part to the client using RTP [6].

6. Media server terminates the SIP session.

It should be noted that the proposed mechanism makes several assumptions about the mobile device, as well as the network services, namely:

+ Mobile device is provisioned with address or addresses of a media server which supports the announcement service defined in RFC 4240 [1]

+ Media Server(s) used by the mobile device support the IMAP URL [4] scheme for the announcement service

+ IMAP Server used by the mobile device supports generating anonymous IMAP URLs using the URLAUTH mechanism

This document assumes that there are no network restrictions between the different components. Specifically it does not address the issues that could occur when the media server and the IMAP server are provided by different entities, and network security protocols restrict the communication between the different components, especially between the media server and the IMAP server.

2.2. Client use of GENURLAUTH Command

The decision to make use of streaming services for a message part will usually be predicated on the content type of the message part. Using the capabilities of the IMAP FETCH command, clients can

determine the MIME [9] Content-Type of particular message parts, and based on local policies or heuristics, decide that streaming for that message part will be attempted.

Once the client has determined that a particular message part requires streaming, the client generates an IMAP URL that refers to the message part according to the method described in RFC 2192 [4]. The client then begins the process of generating an URLAUTH URL, by appending ";EXPIRE=<datetime>" and ";URLAUTH=<access>" to the initial URL.

The ";EXPIRE=<datetime>" parameter is optional, however it SHOULD be used, since the use of anonymous URLAUTH authorized URLs is a security risk, and doing so ensures that at some point in the future, permission to access that URL will cease.

The <access> portion of the URLAUTH parameter SHOULD be 'anonymous'. This document makes the assumption that Media Servers are unlikely to be configured as authorized users to IMAP servers. In this case, without specific prior knowledge of such a configuration, the client MUST use the 'anonymous' access identifier. In the event that the client does have prior knowledge of a media server which is configured for authorized access to the media server, the 'authuser' access identifier SHOULD be used.

The client uses the URL generated as a parameter to the GENAUTHURL command, using the INTERNAL authorization mechanism. The URL returned by a successful response to this command will then be passed to the media server. If no successful response to the GENURLAUTH command is received, then no further action will be possible with respect to streaming media to the client.

Examples:

```
C: a122 UID FETCH 24356 BODY[1.2.MIME]
S: * 26 FETCH (BODY[1.2.MIME] {127}
S: Content-Transfer-Encoding: base64
S: Content-Type: video/mpeg;
S: UID 24356 FLAGS (\Seen))
S: a122 OK FETCH completed.
C: a123 GENURLAUTH "imap://joe@example.com/INBOX/;uid=24356/;\
section=1.2;expire=2006-12-19T16:39:57-08:00;\
urlauth=anonymous" INTERNAL
S: * GENURLAUTH "imap://joe@example.com/INBOX/;uid=24356/;\
section=1.2;expire=2006-12-19T16:39:\57-08:00;\
urlauth=anonymous:\
internal:238234982398239898a9898998798b987s87920"
S: a123 OK GENURLAUTH completed
```

```
C: a122 UID FETCH 24359 BODY[1.2.MIME]
S: * 26 FETCH (BODY[1.3.MIME] {127}
S: Content-Transfer-Encoding: base64
S: Content-Type: audio/G729;
S: UID 24359 FLAGS (\Seen))
S: a122 OK FETCH completed.
C: a123 GENURLAUTH "imap://joe@example.com/INBOX/;uid=24359/;\
section=1.3;expire=2006-12-19T16:39:57-08:00;\
urlauth=anonymous" INTERNAL
S: * GENURLAUTH "imap://joe@example.com/INBOX/;uid=24359/;\
section=1.3;expire=2006-12-20T18:31:\45-08:00;\
urlauth=authuser:\
internal:09823092340928409238409284029348023948 2"
S: a123 OK GENURLAUTH completed
```

2.3. Client Use of the Media Server Announcement Service

Once an authorized IMAP URL has been generated, it is up to the client to pass that URL to a suitable media server that is capable of retrieving the URL via IMAP, and streaming the content to the client using the RTP [6]protocol.

The media server announcement service is used, as described in RFC 4240 [1]. This service allows the client to send a SIP INVITE to a special username ('annc') at the media server (the "announcement" user), supplying the URL obtained as per the previous section.

The SIP INVITE is constructed as shown in the examples below, note that as per RFC 4240, the play parameter is mandatory, and specifies the authorized IMAP URL to be played.

The content-type parameter is optional in RFC 4240, however it MUST be supplied here, using the Content-Type header returned by the IMAP server for the message part. The reason for supplying the content-type parameter is that when the media server issues a URLFETCH command to retrieve the message part, the message part will be returned without any content type information. Since the media server is not likely to have authorized access to other sections in that message, for example the MIME section, then it may fail to stream the content if the content type is not supplied as a parameter to the SIP INVITE URI.

Similarly, the message well be encoded with a content transfer encoding such as base 64. However, RFC 4240 does not include a method for communicating content transfer encoding to the media server as part of the announcement service, nor does the URLFETCH command include a mechanism for retrieving message parts without encoding (c.f. the [BINARY] extension to IMAP). Therefore, an

extension parameter is required, namely a 'content-transfer-encoding' parameter, using the value of the Content-Transfer-Encoding MIME header returned by the IMAP server for the message part. The content-transfer-encoding parameter MUST be supplied if a Content-Transfer-Encoding header for the message part existed in the original message.

Examples of valid SIP INVITE URIs sent to the media server announcement service:

```
sip:annc@ms2.example.net; \
play=imap://joe@example.com/INBOX/;uid=24356/;section=1.2;\
expire=2006-12-19T16:39:\57-08:00;urlauth=anonymous:\
internal:238234982398239898a9898998798b987s87920; \
content-type=video/mpeg; \
content-transfer-encoding=base64
```

```
sip:annc@ms1.example.net; \
play=imap://fred@example.com/INBOX/;uid=24359/;section=1.3;\
expire=2006-12-20T18:31:\45-08:00;urlauth=authuser:\
internal:09823092340928409238409284029348023948 2; \
content-type=audio/G729; \
content-transfer-encoding=base64
```

If the SIP INVITE is successful, as indicated by a 200 OK response, the client can assume that the media server has successfully retrieved the content from the IMAP server, and that the negotiated RTP stream will shortly begin.

An unsuccessful response code of 404 received from the media server indicates that the content could not be found or could not be retrieved for some reason. For example, the media server may not support the use of IMAP URLs. At this point, there are several options to the client, such as using alternate media servers, or giving up in attempting to stream the required message part.

This document does not go into the details of the SIP [5], SDP [11] or RTP [6] protocols; these are well described elsewhere.

2.4. Media Server Use of IMAP Server

A media server receiving a SIP INVITE to the annc user must first attempt to retrieve the content indicated in the play parameter, before returning a response to the client. This section describes how the media server converts the received IMAP URL into suitable IMAP commands for retrieving the content.

The media server first connects to the IMAP server specified in the URL. Once connected, the media server MAY choose to use TLS [10] to encrypt the communication path.

If the media server is configured as an authorized user of the IMAP server, it SHOULD authenticate to the IMAP server using the credentials for that user. This document does not go into the details of IMAP authentication, but the authentication SHOULD NOT use the LOGIN command over a non-encrypted communication path.

If the media server is not configured as an authorized user of the IMAP server, it MUST authenticate to the IMAP server using the LOGIN command, with the username of "anonymous". However, in this case, if the URL supplied in the 'play' parameter of the SIP invite specifies an authorization of 'authuser', then the media server SHOULD NOT attempt to contact the IMAP server, but SHOULD instead immediately return a response code of 404 with a reason phrase of 'Not authorized to access resource'. reason.

Once authenticated, the media server issues the URLFETCH command, using the URL supplied in the 'play' parameter of the SIP invite. A successful URLFETCH command will return the message part, which must be decoded by the media server, according to the content-transfer-encoding header provided by the client (if any). If the URLFETCH was unsuccessful, then the media server MUST return a response code of 404 with an appropriate reason code.

Assuming the content is retrieved and decoded successfully, the media server returns a 200 OK response code, and after an ACK is received, an RTP stream is delivered to the client using the parameters negotiated in the SDP.

The media server MAY choose to implement connection caching, in which case connection and disconnection from the IMAP server are handled according to whatever algorithm the media server chooses.

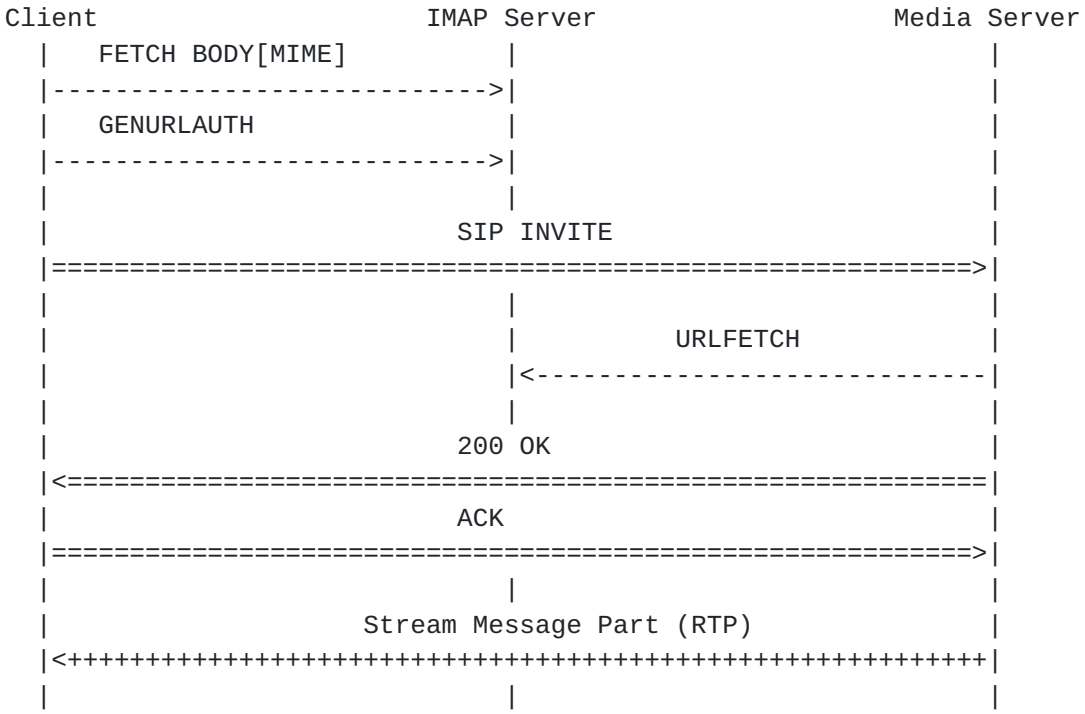
Examples:

```
C: a001 LOGIN anonymous null
S: a001 OK LOGIN completed.
C: a002 URLFETCH
"imap://joe@example.com/INBOX/;uid=24356/;section=1.2; \
expire=2006-12-19T16:39:\57-08:00;urlauth=anonymous:\
internal:238234982398239898a9898998798b987s87920"
S: * URLFETCH "imap://joe@example.com/INBOX/;uid=24356/; \
section=1.2;expire=2006-12-19T16:39:\57-08:00;urlauth=anonymous: \
internal:238234982398239898a9898998798b987s87920" {36}
S: U2kgdmlzIHBhY2VtLCBwYXJhIGJlbGx1bS4K
```

```
S: a002 OK URLFETCH completed.
C: a003 LOGOUT
S: a003 OK LOGOUT completed.
```

2.5. Protocol Diagram

The following diagram shows a simplified view of the protocol interactions between the email client, the IMAP server and the media server. The IMAP, SIP and RTP protocols are distinguished with different line styles.



3. Security Considerations

This document proposes the use of URLAUTH "pawn-tickets" to grant access to message parts that are required to be streamed by the media server. If an anonymous pawn-ticket is used, that grants access to any client of the IMAP server, without requiring any authentication credentials. Even if an authorized pawn-ticket is granted, access is granted to any client that happens to be an authorized user of the IMAP server.

Clearly, any third parties which gain access to the pawn-tickets can potentially access private data. To minimise this, implementors should consider the following:

- IMAP clients SHOULD use encryption such as TLS [10] to secure the communication path to the IMAP server

- The media server SHOULD use encryption such as TLS secure the communication path to the IMAP server

However, the communication path between the client and the media server uses SIP, which currently has no mechanism to secure the data contained in the SIP INVITE URI. Thus, even if an encrypted communication path is used to the IMAP server, the potential for third parties to gain access to the pawn-ticket still exists.

4. Contributors

5. References

[1] Burger, E., "Basic Network Media Services with SIP", RFC 4240, December 2005.

[2] Crispin, M., "Internet Message Access Protocol (IMAP) - URLAUTH Extension", draft-ietf-lemonade-urlauth-08.txt (work in progress), October 2005.

[3] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 3501, March 2003.

[4] Newman, C., "IMAP URL Scheme", RFC 2192, September 1997.

[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[6] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

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

[8] Camarillo, G., "The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)", RFC 3969, BCP 99, December 2004.

[9] Freed, N., Borenstein, N., Moore, K., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME)", RFC 2045, RFC 2046, RFC 2047, RFC 2048, RFC 2049, November 1996.

[10] Dierks, T. and C. Allen, "The TLS Protocol", RFC 2246, January 1999.

[11] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

Author's Address

Neil L Cook
Openwave Systems
e-mail: neil.cook@openwave.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.