Lemonade Mobile
and
New Drafts
Towards Phase 2 of Lemonade

Stéphane H. Maes,
stephane.maes@oracle.com
Ray Cromwell
ray.cromwell@oracle.com

Tasks in plan

- Profile items (possibly) not in profile version 1:
  - Media conversion
  - Quick Reconnect
- Per plan agreed during Lemonade 61.5 interim meeting in Redwood Shores (+ Lemonade 62 discussions):
  - Firewall Traversal
  - Filtering
  - Server to client notifications
  - Transport optimizations (mobile issues)

Starting to address these issues: Proposed new drafts

- draft-maes-lemonade-notifications-server-to-client-01
- draft-maes-lemonade-lconvert-02
- draft-maes-lemonade-lzip-02
- draft-maes-lemonade-ldeliver-02
- draft-maes-lemonade-http-binding-02
- ~~draft-maes-lemonade-monoincuid-01~~

Mobile Concerns and Motivations

- Network overhead
  - Bandwidth consumed and latency
- Client resource consumption
  - CPU, Memory, Radio, Battery Life
- Ease of Deployment
  - For both Operators and Enterprises
- Speed of deployment, Ease of Implementation
  - Handset manufacturing timelines require large lead times
  - Standards track should keep in mind manufacturer product cycles

Server-Side Compose Overview

- Desires:
  - – Compose messages on server with attachments
  - – Reduce bandwidth usage (message download)
  - – Reduce round-trips
  - – Reduce client complexity
  - – Reduce deployment complexity
  - – Support OMA requirements

LCOMPOSE Overview

- LCOMPOSE extends APPEND/CATENATE
- Requires LITERAL+ support
- Adds new literal type: delta encoded literal
- IMAPURL extensions
- Supports OMA Requirement for delta encoding
- New literal support reused by LDELIVER
- Look at optimizing for most common use case

Server-side Composing : Network Overhead Reduction LCOMPOSE vs CATENATE

- CATENATE
  - Extends APPEND to accept multiple TEXT and URL segments
  - Each literal part requires another adds another round trip without LITERAL+
- LCOMPOSE
  - Extends APPEND to accept LITERAL+ and DTEXT literals
  - All inclusions can happen in a single roundtrip
  - Per-inclusion syntax overhead is less (no synchronization)
  - Delta encoded edits may be applied to includes to further reduce network overhead

Server-Side Compose Example CATENATE

```
C: A003 APPEND Drafts (\Seen \Draft $MDNSent) CATENATE (
   URL "/Drafts;UIDVALIDITY=385759045/;UID=20;section=HEADER" TEXT {42}
S: + Ready for literal data
C:
C: --------------030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045/;UID=20;section=1.MIME" URL "/Drafts;UIDVALIDITY=385759045/;UID=20;section=1" TEXT {42}
S: + Ready for literal data
C:
C: --------------030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045/;UID=30" TEXT {44}
S: + Ready for literal data
C: --------------030308070208000400050907–
C: ) S: A003 OK catenate append completed
```

Server-Side Compose Example LCOMPOSE

```
C: A003 APPEND Drafts (\Seen \Draft $MDNSent) CATENATE (
   URL "/Drafts;UIDVALIDITY=385759045/;UID=20;section=HEADER"
    TEXT {42+}
C:
C: --------------030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045/;UID=20;section=1.MIME"
    URL "/Drafts;UIDVALIDITY=385759045/;UID=20;section=1" TEXT
    {42+}
C:
C: --------------030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045/;UID=30" TEXT {44+}
C: --------------030308070208000400050907–
C: )
S: A003 OK APPEND LCOMPOSE completed
```

Server-Side Compose Editing Example

```
C: A003 APPEND Drafts (\Seen \Draft $MDNSent) LCOMPOSE (TEXT {123+}
C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
C: From: Fred Foobar <foobar@Blurdybloop.COM>
C: Subject:         some minor changes
C: To:    mooch@owatagu.siam.edu
C: Content-Type: text/plain; charset=us-ascii
C:
C: DTEXT “/Inbox;UIDVALIDITY=9999/;UID=1234;Section=1” {456+}
C: 3a4,6
C: > The trouble with tribbles is that
C: > they insert themselves in your
C: > files where you least expect it.
C: )
S: A003 OK [APPENDUID 9999 33] APPEND Completed
```

Summary: Composing On the Server

	CATENATE	LCOMPOSE
Roundtrips	min 1 per URL included	1 minimum
Overhead	continuation response	
Complexity	About the same	About the same, except for delta encoded literal support
Features		+ editing / recomposition via delta encoding

Submit of Composed Messages Overview

- Desires:
  - – Reduce bandwidth usage
  - – Reduce round-trips
  - – Reduce client complexity
  - – Reduce deployment complexity
  - – Support almost full SMTP functionality
  - – Design for common cases

LDELIVER

- Allows client to submit Batch SMTP sequences to SMTP server through IMAP
- Using LCOMPOSE extension, pre-composed messages may be included in batch via text literal or delta encoded literal
- Or message may be composed “on the fly” inline, no intermediate draft storage needed

Network Overhead: Submit of Composed Messages

- BURL + SMTP
  - Adds SMTP extension to fetch URLAUTH IMAPURLs as DATA portion
  - Adds 2+ round-trips. One to generate URL, more for SMTP transaction. More verbosity.
  - Requires SMTP server be upgraded to support two standards: URLAUTH and BURL
  - Requires client to utilize URLAUTH
  - Potentially complicates deployment due to simultaneous extensions needed on IMAP and SMTP servers.

Network Overhead: Submit of Composed Messages

- LDELIVER
  - Only one roundtrip needed
  - No special support in SMTP server needed

BURL Example

IMAP SERVER

```
C: a777 GENURLAUTH "imap://joe@example.com/INBOX/;uid=20/ ;section=1.2;urlauth=submit+fred"
      INTERNAL
S: * GENURLAUTH "imap://joe@example.com/INBOX/;uid=20/;section=1.2
      ;urlauth=submit+fred:internal:91354a473744909de610943775f92038"
S: a777 OK GENURLAUTH completed
```

SMTP SERVER

```
C: EHLO potter.example.com
S: 250-owlry.example.com
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-BURL imap
S: 250-AUTH PLAIN
C: AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
C: MAIL FROM:<harry@gryffindor.example.com>
C: RCPT TO:<ron@gryffindor.example.com>
C: BURL "imap://joe@example.com/INBOX/;uid=20/;section=1.2
;urlauth=submit+fred:internal:91354a473744909de610943775f92038” LAST
S: 235 2.7.0 PLAIN authentication successful.
S: 250 2.5.0 Address Ok.
S: 250 2.1.5 ron@gryffindor.example.com
OK. S: 250 2.5.0 Ok.
```

LDELIVER Example

IMAP SERVER

```
C: a123 LDELIVER TEXT {123+}
C: EHLO potter.example.com
C: AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
C: MAIL FROM:<harry@gryffindor.example.com>
C: RCPT TO:<ron@gryffindor.example.com>
C: DATA
C: URL “/INBOX/;uid=20/;section=1.2” TEXT{3+}
C: .
S: * LDELIVER SMTP-RESPONSE {536}
S: 235 2.7.0 PLAIN authentication successful.
S: 250 2.5.0 Address Ok.
S: 250 2.1.5 ron@gryffindor.example.com
S: 250 2.5.0 Ok.
S: a123 LDELIVER completed successfully
```

Sending On the Server

	SMTP/BURL	LDELIVER
Roundtrips	2 minimum setup, (more if PIPELINING not used,+xtra TCP conn)	1 minimum
Overhead	GENURLAUTH + response, BURL cmd	LDELIVER cmd + URL + TEXT
Complexity	Client impl URLAUTH, SMTP+send BURL IMAP Server impl URLAUTH SMTP Server impl BURL/URLFETCH	Client impl LDELIVER, IMAPURL IMAP Server impl LDELIVER, LCOMPOSE LITERAL+, +batch SMTP proxy
Comments		No need to touch SMTP server

Compression

- Mobile clients (GPRS, 1xRTT) are bandwidth constrained
- Mobile bandwidth is expensive
- IMAP is a verbose protocol
- Experiments have shown dramatic compression ratios of IMAP response sequences are achievable

Solutions

- Transport Layer Security (TLS) compression
  - But, not all TLS implementations support compression
  - Deployment of a codec specialized for IMAP may be infeasible
- New IMAP extension LZIP
  - Wraps an IMAP command and indicates to the server to compress all server generated responses using ZLIB
- Defining specialized compression dictionary may be desirable

LZIP Example

C: A001 LZIP A002 FETCH 1:* ALL

S: * LZIP ~{1234}

S: ...[zipped response to FETCH command]...

S: A001 OK LZIP completed

LZIP Compression example ratios 600 messages in INBOX

IMAP Command	Ratio
UID FETCH 1:* FLAGS	3144/20408(15.4%)
SELECT INBOX	249/465 (53.5%)
UID FETCH n BODY[1]	1064/2003 (53.1%)

Compression

	TLS	LZIP
Roundtrips	Minimum 2+	1 minimum
Complexity	TLS/SSL stack + CPU overhead	ZLIB + cpu overhead
Comments	LZIP (defining compression at the application layer) allows some clients to achieve compression without a full SSL/TLS implementation, or where the server does not support the right set of cipher suites, or where an application protocol sensitive codec may be desired	

Server to Client Notifications and Filtering

- Notion of Event-based synchronization
- Server-side filtering
  - Poll and push repository
  - View, notification and event filters
- Inband and outband notifications
- Events
  - Payload
  - SMS binding

Server to Client Notifications

- Support event based synchronization whereby the e-mail server can notify clients of new e-mail and other e-mail server events
- Integrated with IMAP to:
  - – Allow IMAP events to be sent as notifications
  - – Handle delayed or lost notification
  - – Avoid data duplications

Server to Client Notifications and Filtering

- LPROVISION
  - The LPROVISION command is used to allow a device to obtain service specific parameters of the server.
- LSETPREFS and LGETPREFS
  - The LSETPREF command allows a user to define certain configuration parameters, while the LGETPREFS command allows a user to retrieve the configuration values.
- LFILTER:
  - The LFILTER command allows users to name a set of IMAP search terms to be used as a view filters or notification filters, or to get the description or search terms associated with a named filter.

HTTP Binding

- Optional use of HTTP as binding for IMAP
  - This binding is intended to facilitate the use of IMAP in deployments involving a variety of intermediaries
    - offers a standardized alternative to de facto proprietary implementations of such a feature.
  - HTTP allows operators to reuse similar setup and model already used for many other similar and related services, e.g. certain proprietary push e-mail and synchronization offerings, OMA Data Synchronization, Web services and Web access.
  - Using HTTP/HTTPS can simplify deployment in a corporate network through the potential use of a reverse proxy .
  - Also has the advantage of not requiring changes to any firewall configurations and reduces deployment concerns that this often presents to corporation.
    - In general the solution is compatible with any existing firewall.
  - A reverse proxy can also support deployment models that offer roles to other service providers in the value chains, as discussed in OMA Mobiel e-mail AD

HTTP Binding

- HTTP Security, encryption and compression capabilities used with HTTP already implemented in a wide range of existing mobile device can be reused.
- Studies have also shown that a persistent HTTP session has usually proven more resilient than an IMAP IDLE over TCP connection over an unreliable bearer such as a GPRS-based mobile network.
  - Reasons unknown, but speculated to be a shorter time to live given to non-port 80/443 TCP connections
  - Such policy not under control or sphere of IETF
- Use of HTTP as application protocol transport has received much attention[RFC3205]. Concern exists it circumvents firewall policies and misuses or neglects HTTP semantics.
  - draft-maes-lemonade-http-binding-02 does not intent to do so and respect HTTP semantics
  - If suppression of IMAP traffic on HTTP is desired, firewall administrators can still prevent such passage and this can provide incentives to re-configure firewalls to allow solutions on other transports (e.g. TLS) or offer the HTTP-based solution using another provisioned port (e.g. manually, out of band or inband)
  - The ubiquity of HTTP as a transport in most new application protocols both standard and proprietary indicates an underlying political intractability to requiring direct connectivity to non-HTTP ports.
- Aim to allow use of HTTP binding in the widest possible setting, by defining a standard, while enabling firewalls to detect and filter such traffic if they wish to deny such usage

Next Steps

- Collect comments and alternatives
- Moves to WG drafts
- Progress as appropriate