Abstract

Mobile operators usually charge users for the time their mobile client is connected to the Internet and/or for the amount of information sent/received. Thus a mobile client should minimize time it stays connected to its mail server, which suggests that it should disconnect and reconnect frequently. Also, it is possible that the mobile client unexpectedly leaves area of connectivity, which will require that the client reconnects when connectivity returns.

This document defines a quick reconnect IMAP4 extension, which gives a mobile client ability to resume a previously abandoned session, without the need to perform a full synchronization sequence.

Table of Contents

1. Conventions Used in this Document  
2. Introduction and Overview  
3. Session identifier parameter to LOGIN and AUTHENTICATE commands  
   3.1 Extended LOGIN command  
   3.2 Extended AUTHENTICATE command  
   3.3 Session ID parameter to LOGIN/AUTHENTICATE.  
4. SESSION Response  
5. FOLDER Response  
6. RESYNC Response  
7. Formal Syntax  
8. Security Considerations
1. Conventions Used in this Document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

"Resumable session" (or just "session") refers to the entire sequence of client/server interaction from the initial session creation during LOGIN/AUTHENTICATE command until it is explicitly deleted upon client request or until the session expires some time after disconnect. Note, that a resumable session doesn’t terminate when the connection is lost or closed with [IMAP4] LOGOUT command. Also note that this term has a different meaning from the term "session" used by [IMAP4].

<<Editorial comments and questions are enclosed like this>>

<<The term "event" is as defined in ...>>

2. Introduction and Overview

Mobile operators usually charge users for the time their mobile client is connected to the Internet and/or for the amount of information sent/received. Thus a mobile client should minimize time it stays connected to its mail server, which suggests that it should disconnect and reconnect frequently.

Also, it is possible that the mobile client unexpectedly leaves area of connectivity, which will require that the client reconnects when connectivity returns.

IMAP can be verbose. Usually, in order to synchronize a client with a server after a disconnect, the client needs to issue at least the following commands: LOGIN/AUTHENTICATE, SELECT and several FETCH commands (see [IMAP-DISC] for more details).

Thus, there is a desire to have a quick reconnect facility in IMAP, which will give a mobile client ability to resume a previously abandoned session, without the need to perform the full synchronization sequence as described above.

<<Note to RFC editor: This document is compliant with "transitional IMAP capabilities" document [TRANS-CAPA]. Please change the
The quick reconnect IMAP extension is present if an IMAP4 server returns "X-DRAFT-W00-RECONNECT" as one of the supported capabilities to the CAPABILITY command.

<<General description of how this is going to work. The description below assumes the the quick reconnect is mainly used when the client loses TCP connection.

1. The client authenticates and creates a new session by specifying a session ID as an additional parameter to LOGIN/AUTHENTICATE command
2. The client optionally opens a mailbox and does other things
3. At some point the client loses connectivity (or decides to disconnect). The server will retain information about client session (including information about currently selected mailbox) for some amount of time. After that time the session expires.
4. Later on the client reconnects. It authenticates as the same user as before and specifies a previously used session ID. If the session hasn’t expired, it will be resumed, otherwise a new session is created.>>

3. Session identifier parameter to LOGIN and AUTHENTICATE commands

3.1 Extended LOGIN command

Arguments: user name
password
optional Session ID

Data: OPTIONAL untagged responses: SESSION
OPTIONAL - also untagged responses associated with SELECT

Result: OK - login completed, now in authenticated state (or selected) state; see section 3.3
NO - login failure: user name or password rejected
BAD - arguments invalid

This section extends definition of LOGIN command as defined in [IMAP4]. The LOGIN command identifies the client to the server and carries the plaintext password authenticating this user. In addition, the client can pass a session identifier ("SID") in order to attempt to resume a previously created session. The SID parameter is only used if login is successful. If no SID is specified, the LOGIN command behaves as described in [IMAP4]. If a SID is specified, the server first performs all actions associated with a regular LOGIN command. Should authentication succeed, the server must try to resume the specified session as described in section 3.3.

Example: First login, the client needs to perform a state-comparison-sync to get in sync.
   C: A01 LOGIN joe password (SID P6505551234)
   S: A01 OK LOGIN completed

Example: A successful Lemonade login resuming an old session
   C: A02 LOGIN joe password (SID P6505551234)
   S: * SESSION AUTHENTICATED
   S: A02 OK LOGIN completed
Example: A successful Lemonade login resuming an old session in
SELECTED state with the INBOX selected.
C: A02 LOGIN joe password (SID P6505551234)
S: * SESSION SELECTED
S: * FOLDER INBOX
S: * 14 EXISTS
S: * 49 FETCH (....
S: A02 OK LOGIN completed

Example: A successful Lemonade login resuming an old session in
SELECTED state with the INBOX selected, but where the server
could not cache all the events since the last disconnect.
C: A02 LOGIN joe password (SID P6505551234)
S: * SESSION SELECTED
S: * FOLDER INBOX
S: * RESYNC
S: A02 OK LOGIN completed

3.2 Extended AUTHENTICATE command

Arguments: authentication mechanism name
optional Session ID

Responses: continuation data can be requested

Data: OPTIONAL untagged responses: SESSION
OPTIONAL - also untagged responses associated
with SELECT

Result: OK - OK - authenticate completed, now in authenticated
(or selected) state; see section 3.3
NO - authenticate failure: unsupported authentication
BAD - arguments invalid, also authentication exchange
cancelled

This section extends definition of AUTHENTICATE command as defined
in [IMAP4]. The AUTHENTICATE command indicates a [SASL]
authentication mechanism to the server. If the server supports the
requested authentication mechanism, it performs an authentication
protocol exchange to authenticate and identify the client. It MAY
also negotiate an OPTIONAL security layer for subsequent protocol
interactions. If the requested authentication mechanism is not
supported, the server SHOULD reject the AUTHENTICATE command by
sending a tagged NO response. See [IMAP4] for more details.

The client can also pass a session identifier ("SID") in order to
attempt to resume a previously created session. This is only used
if authentication exchange is successful. If it is, than the
presence of a SID tells the server to try to resume the specified
session as described in section 3.3.

<<Check how this will work together with SASL-INITIAL>>

Example: A successful Lemonade authentication exchange resuming
an old session in SELECTED state with the INBOX selected, but
where the server could not cache all the events since the last
disconnect.
C: A02 AUTHENTICATE DIGEST-MD5 (SID P6505551234)
S: + cmVhbG09ImVsd29vZC5pbm5vc29mdC5jb20iLG5vbmbNlPSJPTZN
3.3 Session ID parameter to LOGIN/AUTHENTICATE.

If the session identified by SID is not recognized (for example, this is the first time the client tries to connect, or the session has expired on the server), the server creates a new session and associates it with the provided SID. <<Should the server send some special response for this case? Just reuse RESYNC?>> Otherwise the server resumes a previously created session for the client. For the latter case, the server informs the client of the state of the server by sending an untagged SESSION response. <<Is this really necessary?>> <<Does the client need to know if the mailbox was selected read-only or not?>>

If that state is SELECTED, the server also tells the client what the selected folder is by sending an untagged FOLDER response. Next, the server sends the client any pending events that have occurred in this folder while the client has been disconnected. Thus, the client can just service these pending events and need not perform a full sync. If these events could not be cached for some reason or the server senses the client may have not received some events, the RESYNC Response is returned, and the client should perform a state-comparison based sync. <<Should RESYNC be a response code instead?>>

Whether a session was resumed or a new one was created, the server is now required to remember the session state associated with (username, SID) pair. This state at least includes the currently selected mailbox and an associated state <<EXISTS? RECENT? LASTUID? Other?>>. Other IMAP extensions can define additional state that has to be tracked <<e.g. CONDSTORE>>.

Note, that TLS and/or SASL security layer state [SASL] created with STARTTLS/AUTHENTICATE command on a previous connection can’t be resumed using this mechanism.

<<Define now resuming a session interacts with CONDSTORE, ANNotate and other IMAP extensions>>
Each server MUST be able to keep track of at least 5 sessions for the same user.  
<<What to do if the server ran out of sessions? Allow to carry out work without new session (and add "no more sessions available" response code), or fail LOGIN?>>
Each session’s state MUST be kept by the server for at least 30 minutes after a client disconnect (including LOGOUT command). After that the session is considered expired and its associated state may be deleted.

Note that SIDs are chosen by the client, so they are not unique across the server. The same SID used with two different usernames refers to two different sessions, unless the two usernames refer to the same physical user. SIDs starting from capital letter "P" are reserved for session ids based on international phone numbers.

4. SESSION Response

Contents: session state

The SESSION response returns the current session state <<i.e. whether the user has a mailbox selected or not>>.

5. FOLDER Response

Contents: name of the selected mailbox

The FOLDER response returns the name of the selected mailbox.

6. RESYNC Response

Contents: none

The RESYNC response tells the client that although the specified session has been resumed, the server was unable to cache all events for the client, so the client should perform a full state synchronization.

7. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [ABNF].

Non-terminals referenced but not defined below are as defined by [IMAP4].

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

```
login            = "LOGIN" SP userid SP password [login-params]
authenticate     = "AUTHENTICATE" SP auth-type [login-params] *(CRLF base64)
```
login-params = *(SP "(" login-param-data ")")
    ;; modifies the original IMAP4 login
    ;; command to accept optional parameters

login-param-data = login-param *(SP login-param)

login-param = astring /
    "(" astring SP astring *(SP astring) "")"
    ;; parameters to SELECT may contain one or
    ;; more atoms or strings - multiple items
    ;; are always parenthesized

login-sid = "SID" SP session-id
    ;; conforms to login-param-data syntax

mailbox-data =*/session-resp / folder-resp / resync-resp
    ;; <<Is it Ok to use mailbox-data?>>

session-resp = "SESSION" SP session-state

session-state = ("AUTHENTICATED" / "SELECTED")

folder-resp = "FOLDER" SP folder

resync-resp = "RESYNC"

session-id = string
    ;; a unique session identifier,
    ;; implementation specific. If starts from "P"
    ;; it must be followed by an international
    ;; phone number.

8. Security Considerations

    <<TBD. Mention possibility of Deny of Service, as keeping multiple
    sessions per each user is resource intensive.

    Add a warning about not being able to resume STARTTLS/AUTHENTICATE
    state>>

9. IANA Considerations

    <<TDB>>

10. References

10.1 Normative References

    [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate

    [IMAP4], Crispin, M., "Internet Message Access Protocol - Version

    [ABNF], DRUMS working group, Dave Crocker Editor, "Augmented BNF
    for Syntax Specifications: ABNF", Work in Progress.
[IMAP-DISC] Melnikov, A. "Synchronization Operations For Disconnected Imap4 Clients", work in progress, draft-melnikov-imap-disc-XX.txt


<<[ACL] - informative?, [ANNOTATE?], [CONDSTORE?]>>

10.2 Informative References


11. Acknowledgments

This document borrows some text and ideas from draft-maes-lemonade-command-extensions-00 by S. H. Maes, R. Lima, C. Kuang, R. Cromwell, V. Ha and E. Chiu.

Thanks to <<>> for comments and corrections.

12. Author’s Addresses

Alexey Melnikov
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex, TW12 2BX
UK

Email: Alexey.Melnikov@isode.com

Corby Wilson
Enterprise Mobility Systems
Nokia
503 Martindale Street
Suite 610
Pittsburgh, PA 15212
USA

Email: Corby.Wilson@nokia.com

13. Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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


<<Note that this section will be deleted before publication>>

15.1 Change Log

00 Initial Revision.

15.2 Open Issues for Discussion

15.3 To Do