IMAP (Internet Message Access Protocol)
IMAP4rev1 [RFC2060]

情報工学専攻 岡村研究室 修士一年 開内大介

平成 14 年 3 月 27 日

1 はじめに

社会へのインターネットの普及に伴ってその利用が増大し，そのネットワーク上におけるサービスも多用化してきている．このサービスとして一般的なものは，Web ページや電子メール，動画や音楽などのマルチメディアコンテンツなどが挙げられる．このようなサービスでは送受信者間で RFC によって定められた規約 (プロトコル) を守ることによって通信を行なうことが重要である．これは一般的なサービスとして先に挙げた電子メールも同様であり，RFC で定められているプロトコルとしては POP3(Post Office Protocol - Version3) や SMTP(Simple Mail Transfer Protocol)，IMAP が挙げられる．このうち SMTP は電子メールの送信に関する通信を定めたプロトコルである．これに対して，POP,IMAP はともに電子メールの受信に関する通信を定めたプロトコルとなっている．本レポートではこの電子メールを扱うプロトコルの一つである IMAP を紹介する．

2 POP と IMAP について

POP とは Post Office Protocol の略で，現在では POP3(POP version3) が主に利用されている．また，この POP3 は RFC1939 で定義されている．このプロトコルはメールサーバに到着した電子メールを PC 等のクライアントに取り込むためのプロトコルである．つまり，ユーザ側で電子メールの操作をする場合には必ず PC などのクライアント側にメールをダウンロードした後，クライアント側で操作することが前提である．このプロトコルで利用できる機能としては次のようなものが挙げられる．

- メールの取得
- メールの削除
- メールスプールに蓄積されているメール数の取得
- メールの情報の取得

次に，IMAP について説明する．IMAP とは (Internet Message Access Protocol) の略で，現在は主に RFC2060 で定義されている IMAP4rev1(IMAP version4 revision1) が使用されている．IMAP は後に挙げるような POP3 で考えられる問題点を改善するために考えられたプロトコルと言える．そのため POP ではクライアントはメールサーバからメールをクライアント側にダウンロードすることを前提としているが，IMAP ではメールに対する操作をメールサーバ上で行なうことを前提としている．IMAP4rev1 の機能としては次のようなものが挙げられる．

- メールの取得
- メールの削除
- メールの情報の取得
- メールスプールに蓄積されているメール数の取得
- メールフォルダに蓄積されているメール数の取得
- メールフォルダの作成，削除
- メールの整理，検索

次に，POP の問題点とそれに対する IMAP の利点について述べる．POP は現代の状況で用いるには適していないという問題点が挙げられる．現代ではモバイル環境が整備されつつあり，一人が複数のクライアントを用いて同じメールボックス内のメールを読むという状況が考えられる．ここでこのような状況において POP を用いることを考える．まず，自宅のパソコンにある POP クライアントでメールサーバからメールをダウンロードしてメールを読んだとする (これは POP では先に述べたようにクライアントがメールを

ダウンロードすることを前提としているため) . このあと外出先で同じメールサーバにある自分のメールを読もうとする場合 , 新着メールのダウンロードはできるが自宅パソコンで見ることができるようなメールの状態を再現するということはできない .

ここで , 同じ状況において IMAP を利用することを考えると , このような状況はなくなる . まず , 自宅パソコンから IMAP クライアントを用いてメールサーバに接続すると , そのメールサーバ上でメールの読み書きや整理などを行なう . その後外出先で IMAP クライアントを用いてメールサーバに接続すると , 自宅で見たメールボックスの状況が再現される . つまり , メールサーバ上でメールに対する操作を行なうことによって , 複数のクライアントからメールを読む状況に適した環境を作り出すことができる .

また , 計算機資源に関していうと IMAP は現代のモバイル環境により適したプロトコルであるといえる . これは POP はメールデータをクライアント側にダウンロードしてメールに対する操作 (整理 , 検索など) を行なう . つまり , この場合クライアント側に計算機資源を必要とする . これに対して , IMAP ではメールサーバ上でメールに対する操作を行なうため計算機資源はメールサーバ側に要求される . よって , モバイル環境においては IMAP の方がより好ましいプロトコルであると言える .

3 IMAP4rev1 の機能

ここで IMAP4rev1 の細かい機能について述べる . IMAP4rev1 でサポートされている機能としては次のようなものが挙げられる . これらの機能はクライアントがメールサーバと接続されている状況で行なわれるものだが , IMAP ではメールサーバとの接続が切れている状況でのメール操作についても考慮されている . しかし , 現時点では厳密な規定がされていないため , クライアントソフトウェアによって動作が異なる .

- mailbox の作成や削除 , 名前変更
- 新着メールのチェック
- メッセージの削除
- メッセージのフラグの設定変更
- (宛先 , 日付 , 文字列 , メッセージ ID などによる) メールの検索
- 選択したメールの取り込み

4 IMAP4rev1 のプロトコル説明

本章では IMAP4rev1 のプロトコルに関する説明を行なう . 以後 , IMAP クライアントは IMAP が利用可能なクライアント , また IMAP サーバは IMAP が利用可能なメールサーバを意味する .

4.1 IMAP4rev1 の状態遷移について

IMAP4rev1 のプロトコルは状態を持つ . それぞれの状態によってそのときに実行できるコマンドが制限される . ここでいうコマンドとは , 後に説明する IMAP プロトコルで定義されているコマンドを意味する . IMAP4rev1 で定義されている状態は次の 3 つがある . また , これらの状態に関する状態遷移が図 1 である .

- Non-Authenticated state

  IMAP クライアントと IMAP サーバ間でユーザ認証を行なう前の状態 .

- Authenticated state

  IMAP クライアントと IMAP サーバ間でユーザ認証が行なわれた後の状態 . メールボックスの操作が可能 .

- Selected state

  メールボックスを選択した状態 . メールへの操作が可能 .

4.2 IMAP4rev1 のサーバ応答について

IMAP4rev1 のプロトコルで定義されているサーバの応答について説明する . IMAP4rev1 のプロトコルでは送受信されるメッセージの始めにはタグを付けたメッセージとタグをつけないメッセージの 2 種類が存在する . タグは大抵アルファベットと数字によって構成される文字列であり , IMAP のあるコマンドとそれに対する応答の二つのメッセージは同じタグを用いる .

コマンド応答例

```
C->S: 001 CAPABILITY
S->C: * CAPABILITY IMAP4 IMAP4rev1 AUTH=LOGIN
S->C: 001 OK CAPABILITY completed
```

- タグなし応答
  コマンド実行に付随して発生した情報の伝達用
  コマンド応答例では”001”

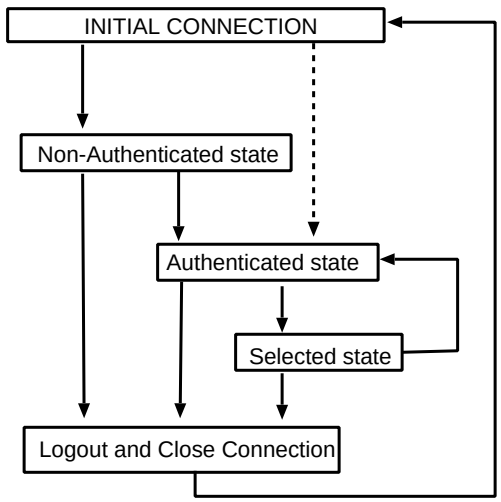

図 1: IMAP4rev1 の状態遷移図

- タグ付き応答
  コマンドの実行結果の通知用
  コマンド応答例では” ∗ ”

4.3 IMAP4rev1 のコマンドについて

本章では，IMAP4rev1 のプロトコルで定義されているコマンド群に関して説明する．IMAP のプロトコルでは，基本的にタグから始まり，コマンド名，その引数という文法となっている．

4.3.1 すべての状態で使用可能なコマンド

- CAPABILITY : サーバの能力一覧を取得
- NOOP : 何もしない
- LOGOUT : ログアウト

ここで述べるコマンドは 4.1 章で述べたすべての状態で利用可能であるコマンドである．CAPABILITY コマンドは IMAP プロトコルの最初に使用するコマンドであり，このサーバ応答にはそのサーバで利用できる IMAP のバージョンや認証方法などが含まれる．NOOP コマンドはそれ自信は特に意味を持たないが，このサーバ応答で新着メールの有無をチェックしたり，またクライアントが一定時間無応答の場合に自動的にログアウトされてしまうためのタイマーのリセットなどの役割を持つ．LOGOUT コマンドはサーバからログアウトするコマンドである．

4.3.2 Non-Authenticated state で使用可能なコマンド

- LOGIN : ユーザ名と平文パスワードで認証
- AUTHENTICATE : 引数で指定した方法で認証する (LOGIN, CRAM-MD5, KERBEROS_V4)

ここでは Non-Authenticated(認証前) の状態で使用できるコマンドを紹介する．LOGIN コマンドはユーザ名と平文パスワードをサーバ側に送ることでサーバとクライアント間で認証を行なうコマンドである．AUTHENTICATE コマンドは引数で指定した認証方式を用いて認証を行なうことができる．これにより，LOGIN よりも安全な認証を行なうことができる．(認証方式の種類については上記を参照)

4.3.3 Authenticated state で使用可能なコマンド

- SELECT : メールボックスの選択
- EXAMINE : メールボックスの選択 (read-only)
- CREATE : メールボックスの作成
- DELETE : メールボックスの削除
- RENAME : メールボックスの名前変更
- SUBSCRIBE : メールボックスの購読
- UNSUBSCRIBE : メールボックスの購読中止
- LIST : メールボックス一覧の取得
- LSUB : 購読メールボックス一覧の取得
- STATUS : メールボックスの状態取得
- APPEND : メールボックスにメールを追加

ここで挙げられているコマンドは Authenticated(認証後) の状態で使用可能なコマンドである．これらはメールボックスに対する操作を行なうコマンドとなっている．SELECT コマンドは指定したメールボックスの選択を行なうコマンドであり，EXAMINE コマンドは read-only の状態でメールボックスの選択を行なう．CREATE, DELETE, RENAME コマンドはそれぞれメールボックスの作成，削除，名前変更を行なうコマンドである．

SUBSCRIBE コマンドは指定したメールボックスをそのユーザが通常使用するメールボックスとして登録するというコマンドである．UNSUBSCRIBE コマンドは SUBSCRIBE コマンドとは逆に購読の中止を行なうことができる．LIST コマンドはそのクライアントが使用可能なメールボックスの一覧を得るためのコマ

ンドである．このとき，同時に各メールボックスの属性も得ることができる．メールボックスの属性は以下の 4 つが存在する．LSUB コマンドでは SUBSCRIBE コマンドで購読するように設定したメールボックス一覧を得ることができる．

LIST コマンドで得られるメールボックスの属性

- \NoInferiors　下層のメールボックスを作成できない
- \Noselect　選択不可能
- \Marked　メールが追加されている
- \Unmarked　メールが追加されていない

4.3.4 Selected state で使用可能なコマンド

- CHECK　：メールボックスの状態等のチェック
- CLOSE　：メールボックス選択解除
- EXPUNGE　：メール削除実行
- COPY　：メールのコピー
- UID　：UID を使用したコマンド実行
- SEARCH　：メールの検索
- FETCH　：メールの取得
- STORE　：メールフラグの変更

ここで上げられているコマンドは Selected 状態，つまりメールボックスを選択した後の状態であるときに実行できるコマンド群である．これらは主にメールに対する操作のためのコマンドとなっている．

CHECK コマンドはメールボックスの状態などののチェックと表記している．これは RFC ではサーバの実装に依存したサーバ側での作業の実行を行なうと定義されている．例えばメモリ上でのメールボックスの状態とディスク上でのメールボックスの状態の比較などがこれにあたる．しかし，サーバ側で特にこのような作業が必要でない場合では，CKECK コマンドは NOOP コマンドと等価なコマンドとなる．

次に CLOSE コマンドはそのときのメールボックスの選択を解除するコマンドである．これを実行することによって Authenticated(認証後) の状態に状態遷移することになる．

EXPUNGE コマンドと COPY コマンドはそれぞれ指定したメールデータの削除と複製を行なう．

次に，UID コマンドはメールそれぞれに対して一意につけられる UID(Unique IDentifier number) を使用して，そのメールデータに対して COPY, FETCH, STORE, SEARCH コマンドを実行することができる．

SEARCH コマンドは現在選択しているメールボックス内のメールデータに対する検索を行なうことができる．この検索時に利用できる検索キーとしては下記に上げているように多くのものが利用できる．これらの検索キーを使用することによって柔軟な検索を行なうことができる．IMAP ではプロトコル内で検索を行なうコマンドが定義されているため，実装に依存しない検索を行なうことができる．

SEARCH コマンドで使用可能な検索キーの一覧
(斜体文字は文字列の示す任意のデータ)

• メッセージ ID を使用する検索キー *message-seqno-set*, ALL, UID *uid-set*
• 宛先に関連する検索キー To *address*, From *address*, CC *address*, BCC *address*
• 日付けに関する検索キー BEFORE *date*, ON *date*, SINCE *date*, SENTBEFORE *date*, SENTON *date*, SENTSINCE *date*
• 含まれる文字列に関連する検索キー HEADER *field value*, BODY *string*, SUBJECT *string*, TEXT *string*
• メール長に関連する検索キー LARGER *length*, SMALLER *length*
• メールのフラグに関連する検索キー RECENT, NEW, OLD, ANSWERED, DELETED, DRAFT, FLAGGED, SEEN, KEYWORD *string*, UNANSWERED, UNDELETED, UNDRAFT, UNFLAGGED, UNSEEN, UNKEYWORD *string*
• 検索キーの演算を行なうための検索キー NOT *search-key*, OR *search-key1 search-key2*

(message-seqno-set は各メールに割り当てられる各メールボックス内で一意な番号である Message Sequence Number の集合,uid-set は UID の集合を意味する)

FETCH コマンドは指定したメールを受信するコマンドである．このコマンドを実行する際に取得キーを利用することによってメッセージの一部分のみ (メールデータのヘッダやメール本文) といったメールデータの部分的な取得が可能となっている．

FETCH コマンドで使用可能な取得キーの一覧
(斜体文字は文字列の示す任意のデータ)

• BODY[section].<region>
• BODY.PEEK[section].<region>
• BODY メール本文の構造
• BODYSTRUCTURE BODY と同じだが，拡張ヘッダも含む
• ENVELOPE メールのエンベロープの構造
• FLAGS メールに付けられたフラグの情報
• INTERNALDATE メールの内部日付
• UID メールの UID
• RFC822 BODY[] と機能的に等価だが，その結果のフォーマットが異なる
• RFC822.SIZE RFC822 のメッセージのサイズ
• RFC822.HEADER BODY.PEEK[HEADER] と機能的に等価だが，その結果のフォーマットが異なる
• RFC822.TEXT BODY[TEXT] と機能的に等価だが，その結果のフォーマットが異なる
• ALL FLAGS INTERNALDATE RFC822.SIZE ENVELOPE を指定した場合と等価
• FAST FLAGS INTERNALDATE RFC822.SIZE を指定した場合と等価
• FULL FLAGS INTERNALDATE RFC822.SIZE ENVELOPE BODY を指定した場合と等価

次に STORE コマンドについて述べる．これは各メールに対して付けられるフラグを操作するためのコマンドである．メールに対して付けられるフラグの一覧は下記のように 6 つ存在する．このフラグの中で，Recent フラグ (新着メールに付けられるフラグ) だけは STORE コマンドで操作できない．

IMAP4rev1 で使用されるメールのフラグ
\ Seen : 既読
\ Answered : 返信済み
\ Flagged : 重要，緊急
\ Deleted : 削除予定
\ Draft : 書きかけ
\ Recent : 新着 (STORE コマンドで設定不可)

5 IMAP4rev1 クライアント

IMAP クライアントソフトウェアは POP3 対応のクライアントソフトウェアの数と比べるとまだ数は少ないといえる．しかし，近年完成度の高い IMAP 対応のクライアントソフトウェアが開発されてきている．ここではその代表として Sylpheed を紹介する．Sylpheed は山本 博之さんが作っている GTK+を使用した X Window System 上で動作する電子メールクライアント&ニュースリーダーである．Sylpheed の対応しているプロトコルとしては APOP，POP3,IMAP4rev1, SMTP, SMTP AUTH, NNTP がある．また，POP3, SMTP, IMAP4rev1 での SSL を用いた通信もサポートしている．この IMAP と POP3 における SSL を用いた通信については RFC2595 で定義されている．また，SMTP での SSL を用いた通信については RFC2487 で定義されている．このような SSL 通信を用いることによってより安全なメール操作が可能となる．

6 まとめ

本稿では IMAP の紹介として，まず従来用いられてきたメール受信用のプロトコルである POP との比較を行なった．次に現在の IMAP プロトコルの主流である IMAP4rev1 の機能とそのプロトコルで定義されているコマンドを紹介した．

先に述べたように IMAP のプロトコルは，現代環境において POP3 を利用する上で考えられる問題点を改善したプロトコルであるといえる．しかし，まだ IMAP プロトコルは完成してはおらず多くの研究者の手で現在も改良が進められている状況である．今後は

多くの人にとってより便利なプロトコルとなることを
期待したい．

参考文献

[1] INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1 (RFC2060)
http://www.ietf.org/rfc/rfc2060.txt?number=2060

[2] Post Office Protocol - Version 3 (RFC1939)
http://www.ietf.org/rfc/rfc1939.txt?number=1939

[3] Using TLS with IMAP, POP3 and ACAP (RFC2595)
http://www.ietf.org/rfc/rfc2595.txt?number=2595

[4] SMTP Service Extension for Secure SMTP over TLS (RFC2487)
http://www.ietf.org/rfc/rfc2487.txt?number=2487

[5] Sylpheed (IMAP 対応の電子メールクライアント)
http://sylpheed.good-day.net/