RFC1459
Internet Relay Chat

情報工学専攻 程研究室
花田 泰紀

平成 11 年 2 月 8 日

1 IRC の概要

IRC(Internet Relay Chat) プロトコルとは、テキストベースでの会話を行うためのプロトコルのことで、その構成要素はサーバ/クライアント構造で構成されている。また、サーバ/クライアントで構成される IRC ネットワーク内には、"チャンネル" と "オペレータ" という概念が存在する。それぞれ、チャンネルは会話を行うグループであり、オペレータはチャンネルや IRC ネットワークを管理する権限のことである。

1.1 サーバとクライアント

サーバは IRC のバックボーンを形成しており、クライアントが IRC ネットワークに接続する端点を提供している。また、サーバ間の接続には、木構造 (spanning tree) のみが許されている。

IRC では、サーバに接続するサーバ以外の全てはクライアントとして扱われ、それぞれのクライアントは 9 文字までのニックネームで識別される。その時サーバは、接続してきたクライアントに関する以下の情報を保持しなければならない。

(a) クライアントの走っているホスト名
(b) そのホストでのクライアントのユーザ名
(c) クライアントの接続しているサーバ名

1.2 オペレータ

IRC には、IRC ネットワークを管理する "IRC オペレータ" と、チャンネルを管理する "チャンネルオペレータ" の 2 つのオペレータが存在する。

IRC オペレータは、IRC ネットワークを管理することができ、IRC ネットワークの高レベルの保守機能を使用することができる。その一方で、与えられた権限は危険であるかもしれないが、IRC ネットワークには必要な存在となっている。

チャンネルオペレータ (`@nickname` で表示) は、IRC のチャンネルを管理することができ、以下の 4 つの権限を行使することができる。

`KICK` - チャンネルからクライアントを追放
`MODE` - チャンネルモードを変更
`INVITE` - クライアントをチャンネルに招待
`TOPIC` - チャンネルのトピックを設定

2 チャンネル

チャンネルとは、1 つ以上のクライアントで構成された会話のグループのことを言う。その「チャンネル」に宛てられた全てのメッセージは、チャンネルを構成するクライアントに送られる。また、チャンネルは最初のクライアントが参加したときに生成され、最後のクライアントが抜ける時に消滅する。

2.1 チャンネル名

チャンネル名は、'&' か '#' で初まる 200 文字までの文字列を利用することができるが、' '(空白)、C-g、','の各文字を使用することはできない。また、チャンネルの頭文字によって、チャンネル宛てのメッセージの配送が以下のように制限される。

チャンネル名の頭文字	配送範囲
'&'	クライアントが接続しているサーバのみに配送
'#'	IRC ネットワークのすべてのサーバに配送

表 1: チャンネル名の 1 文字目とその意味

さらにチャンネル名には、#foo:*.jp のように':' の後にサーバのマスクを付加することができ、一部の実装では、%foo というチャンネル名で#foo:*.jp を表すものがある。

2.2 チャンネルモード

チャンネルには以下の属性があり、チャンネルオペレータによってその値を変更することができる。

o - チャンネルオペレータの権限の付与/剥奪
p - プライベートチャンネル属性の設定/解除
i - インバイトオンリー属性の設定/解除
t - トピックの変更権をチャンネルオペレータのみに設定/解除
n - チャンネル外のクライアントからのメッセージ受信を許可/拒否
m - moderated チャンネル属性を設定/解除
l - チャンネルの参加人数を制限 (<limit>)
b - チャンネルへの参加を拒否する Ban マスクの設定/解除
v - moderated チャンネルでの発言権の付与/剥奪
k - チャンネルキーの設定

表 2: チャンネルモード

MODE メッセージで設定 (メッセージについては、3 章を参照)
MODE メッセージの形式は以下の通り。

```
MODE <channel> {[+|-]|o|p|s|i|t|n|b|v|} [<limit>] [<user>] [<banmask>]
```

3 メッセージ

サーバ/クライアント間の通信は全てメッセージの送受信で行われる

3.1 メッセージ形式

IRC での通信は、テキストベースの “メッセージ” の送信によって行われる。メッセージは以下の 3 つの要素で構成されている。

(a) プレフィクス (サーバ-サーバ間のメッセージだけに用いる)
(b) コマンド
(c) コマンドパラメータ (最大 15 個まで)

表 3: メッセージの構成要素

詳細なメッセージの形式は以下の擬似 BNF によって定義されている。ここで、`<trailing>`はメッセージ中に空白文字を許すためのトリックとなっている。

```
<message> ::= [':' <prefix> <SPACE>] <command> <param> <crlf>
<prefix>  ::= <servername> | <nick> [ '!' <user> ] [ '@' <host>]
<command> ::= <letter> {<letter>} | <number> <number> <number>
<SPACE>   ::= ' ' {' '}
<params>  ::= <SPACE> [ ':' <trailing> | <middle> <params> ]
<middle>  ::= <先頭が':' でない SPACE,NUL,CR,LF を含まない、非空のオクテット列>
<trailing>::= <NUL,CR,LF を含まないオクテットの列>
<crlf>    ::= CR LF
```

3.2 各種メッセージ

メッセージは以下のものがあり、これで全てである。

接続の開始/終了	– PASS, NICK, USER, SERVER, OPER, QUIT, SQUIT
チャンネルの操作	– JOIN, PART, MODE, TOPIC, NAMES, INVITE, KICK
サーバへのクエリー	– VERSION, STATS, LINK, TIME, CONNECT, TRACE, ADMIN
メッセージの送信	– PRIVMSG, NOTICE
ユーザ情報の取得	– WHO, WHOIS, WHOWAS
その他	– KILL, PING, PONG, ERROR
オプショナルメッセージ	– AWAY, REHASH, RESTART, SUMMON, USERS, WALLOPS, USERHOST, ISON

表 4: 各種メッセージ

3.3　ニュメリックリプライ

メッセージへの返信はニュメリックリプライとして返される (一部のメッセージでは返されない)。ニュメリックリプライには大きく分けて、エラーリプライとメッセージ応答があり、それぞれエラー時と正常時のリプライとなっている。ニュメリックリプライは 3 文字の数字文字列からなっている。

4　メッセージの送信

IRC では、いくつかのメッセージ送信の形態がある。いずれにおいても、それらは最小限の配送になるような仕組みになっている。

4.1　クライアント → クライアント

送信先のクライアントの接続されているサーバまでの最短経路を通り送信される。

4.2　クライアント → 複数のクライアント

複数へのクライアントへの送信には、リストを用いるものと、チャンネルを用いるものがある。

リストを用いる場合には、リストはサーバで解釈されて、それぞれの宛先にメッセージを送信する。この場合メッセージが複数送信されることになるので効率が悪い。一般的には次に挙げるチャンネルを用いた送信を用いている。

チャンネルを用いる時には、チャンネルに参加しているクライアントがいるサーバにのみメッセージを送信し、メッセージを受け取ったサーバは、IRC ネットワーク内のそのチャンネルのクライアントを持つサーバにメッセージを送信し、受け取ったサーバはそのサーバに接続しているそのチャンネルに参加しているクライアントにメッセージを送信する。サーバネットワーク内で、チャンネルに関する情報を交換しあっているので、最低限の配送を行うことができるようになっている。

4.3　クライアント /サーバ→ 全体

全体へのメッセージは IRC ネットワークへのブロードキャストと言うことができ、大量のトラヒックを生じるが、いくつかのメッセージはこの方法でのみ送信することができるようになっている。

クライアントからサーバ全体へ送信は、大くの状態情報 (チャンネルメンバー、チャンネルモード、ユーザ情報など) が送信される。

サーバからサーバ全体への送信は IRC の根幹であり、サーバから送信されるメッセージのほとんどは、他の全サーバへのブロードキャストとなっている。

なお、クライアントから全てのクライアントへのメッセージは用意されていない。

5　実際の接続

5.1　サーバへの接続

サーバへの接続は以下の 3 つのメッセージによって行われる。

PASS：パスワードを送信 (オプショナル)

IRC では、クライアントのパスワードを設定することができ、接続の最初に送ることで使用可能となる。最後に送られたパスワードが有効となり、何度送ってもよいことになっている。また、パスワードは設定しなくても IRC を利用することはできる。PASS メッセージの形式は以下の通り。

```
PASS <password>
```

NICK：ニックネームを設定

IRC では、クライアントの識別にニックネームを利用している。そのため PASS メッセージを送信しなければらなない。ニックネームに衝突が無ければ設定は完了。ニックネームが衝突する時には再設定する必要がある。NICK メッセージの形式は以下の通り。

```
NICK <nickname> [ <hopcount> ]
```

USER：ユーザ情報を送信

NICK メッセージと USER メッセージが正しく送信されて初めて接続が完了する。USER メッセージの形式は以下の通り。

```
USER <username> <hostname> <servername> <realname>
```

irchat でのサーバへの接続

irchat での接続は、M-x irchat で行う。その際、サーバが指定されていない場合には、どのサーバに接続するのかを聞かれるようになっている。接続するサーバとポートは、以下のように設定することができる。

```
(setq irchat-server "irc.karrn.ad.jp")
(setq irchat-service 6667)
```

5.2 チャンネルへの JOIN

IRC でチャンネルに参加する時には、JOIN メッセージを利用する。チャンネルへの JOIN が成功するには、以下の 3 つの条件を満たしていなければならない。

1. そのチャンネルがインバイトオンリーチャンネルでない (インバイトオンリーチャンネルの時は、チャンネルオペレータによる INVITE によってのみ参加可能)。
2. クライアントが Ban リストにマッチしてはならない
3. チャンネルキーが設定されているときは、キーが必要

JOIN メッセージの形式は以下の通り。

```
JOIN <channel> {,<channel>} [<key>{,<key>}]
```

irchat でのチャンネルへの JOIN

irchat でのチャンネルへの JOIN は、C-c j (irchat-Command-join) で行う。その際に JOIN するチャンネル名を聞かれるようになっている。また、サーバに接続する時に、自動的に JOIN するチャンネルを以下のように設定することができる。

```
(setq irchat-startup-channel-list '("#foo:*.jp"))
```

5.3 IRC での発言

IRC での発言は PRIVMSG メッセージの送信の送信によって行われる。発言は特定のクライアントやチャンネルに対して行われる。

PRIVMSG メッセージの形式は以下の通り。

```
PRIVMSG <receiver> {,<receiver>} <text>
```

irchat での発言

irchat での発言は、RET (irchat-Command-send-message) によって行う。*IRC* バッファ中の現在カーソルのある行の内容が、現在着目しているチャンネルに送信される。

5.4 チャンネルからの PART

IRC でチャンネルから離脱する時には、PART メッセージを利用する。

PART メッセージの形式は以下の通り。

```
PART <channel> {,<channel>}
```

irchat でのチャンネルからの離脱

irchat でのチャンネルからの離脱は、C-c C-p (irchat-Command-part) によって行う。現在着目しているチャンネルからの離脱を行う。

5.5 サーバからの切断

IRC でのサーバからの切断は、QUIT メッセージによって行われる。QUIT メッセージの形式は以下の通り。

```
QUIT {<quit message>}
```

irchat でのサーバからの切断

irchat でのサーバからの切断は、C-c q (irchat-Command-quit) によって行う。また、サーバと切断すると同時に irchat も終了する。