

第 18 部

IRC の運用技術と活用技術

第 1 章

はじめに

IRC WG は、運用技術と活用技術の研究を中心に IRC (Internet Relay Chat) に関する諸問題を取り扱ってそれら技術の研究開発を目標にするグループであり 1998 年 10 月に設立された。

一方、それ以前からこの WG のメンバーによって IRC に関する様々な研究と活動を行なってきており、古くは IRC の普及から日本における IRC の発展を支えてきている。

IRC は 1988 年にフィンランドで開発されてから既に 10 年が経ち、日本においても現在も年間数倍の勢いで利用者数が増えているが、他の古くからあるコミュニケーションメディアであるメールやネットニュースやウェブなどと比較してメジャーであるとはいえない。

そこで本報告書では、ここ 10 年近くの活動およびまとめとして、まず IRC の概要と特徴などを論じ、次に日本における活動や開発と日本語における問題を、そして最後に IRC サーバ運用管理に関する技術や諸問題と解決について報告する。

第 2 章

IRC について

2.1 IRC の性質

IRC (Internet Relay Chat) はインターネット上でのテキストベースの遠隔会議システムであり、1988 年にフィンランドで開発されて現在も様々な改良を続けながら世界中へと広まっている。IRC のシステムは、複数の IRC サーバおよび複数の IRC クライアントからなり、単体のサーバ、あるいは、複数のサーバが接続された IRC 網に対して、各ユーザはクライアントとして最寄りのサーバへ接続することによって利用する。同じ IRC 網の上の異なるサーバに接続したユーザ同士の会話は、IRC 網の中で複数のサーバ上をリレーされることによってお互いに伝えられる。

TCP/IP ベースのインターネットを生かし、遠くはなれたもの同士がリアルタイムに双方向の会話を行なうアプリケーションとして talk や phone が古くから利用されてきたが、会話を行なう相手がどのマシンを利用しているかを調べそれぞれ相手を指定して呼び出してやらないといけないことや、同時に会話に参加できる人数に制限があることなど、利用にあたって多くの問題があった。

また、メーリングリストやネットニュースあるいはウェブの掲示版などではリアルタイムに会話をすることが困難であるとともに、会話のための一行に対して交換されるデータに付随するヘッダデータが大きくなるなど非効率である。また、ユーザインタフェースもそのような使い方ではなくてあるまとまった大きさのメッセージ群を期待して作られているのがほとんどである。

一方、ウェブ上において CGI などによって作られたチャットシステムもあるが、会話のための一行に対して交換されるデータに付随するヘッダデータが多だけでなく、他の人の会話が来っていないかを調べるために定期的に毎回サーバへ接続してチャット画面を全部取得し、サーバやネットワークへの負荷も高いといった問題があり、さらにユーザインタフェースもかなりつらいものになっている。

これらに対し、IRC はインターネット上でのテキストベースの遠隔会議システムとしてはじめから設計されており、それらの問題を解決している。IRC は同時に多人数の相手とコミュニケーションができるだけでなく以下のような特徴や利用方法を備えている。

- 即時性 / 双方向性

メールやニュースと違って IRC の最大の特徴は、リアルタイムなメッセージ交換ができることにある。インターネット上で距離を感じることなく気軽におしゃべりを楽しんだり会議など打ち合せをしたりすることによって、仮想的なコミュニケーションの場がより一層感じることができる。

メールで短時間に短いフォローを何通もやりとりしているのをよく見かけるが、IRC を使えば対話的に効率よくコミュニケーションすることができる。また、相手がお互い目の前にいるような感じで議論も進みやすいため、現実に出て話しをする時と同様に議論でき、ネットニュースなどでよく起こるようなすれ違いやあげあしとりの批判を繰り返すようなことなどは起きにくい。

- 非同期性

talk や phone では、会話をはじめるにあたって相手がどこにいるか調べてそれぞれ呼び出してやる必要があり、あとから加わろうとするものも同様に呼び出してやらなければならない。これに対し、IRC では誰もがいつでも好きな時に会話に参加したり会話から抜け出したりする事が可能である。

また、お互いそれぞれ常に IRC に参加したままにいることによって、そのとき席を離れている相手へも伝言を残しておける。その場合、数時間後に返事が来たりする場合もありメールやニュースと同じく相手を束縛せず非同期にコミュニケーションをすることも可能である。もちろん、そのようなやりとりの途中で同時に複数人が居合わせれば、すぐにもリアルタイムなメッセージ交換を始めることができる。

- 放送性

IRC のチャンネルは生でリアルタイムな種々の情報やデータなどを流すミニ情報放送局として使われることもある。オープンな設定のチャンネルでは出入りが自由なので、その放送を聴きたい人が聴きたい時にそのチャンネルへ入ればよい。IRC ではチャンネルで発言するのを特定の者だけに制限する設定もでき、このような情報放送や公開会議を行なう際には聴衆からの雑音を禁ずることもできる。

2.2 IRC における基本概念

2.2.1 ニックネーム

IRC においては、多数のユーザを識別するものとしてニックネームという概念を用いる。各ユーザはそれぞれ異なるニックネームを持ち、一つの IRC サーバ網内ではそれぞれの時点で唯一のものとなるため、お互いに相手を識別したり指定したりするのに使われる。例

えば、相手のニックネームを指定してプライベートメッセージを相互に送り合うことで、二人のみの閉じた会話を行うことができる。

ニックネームとして用いることができる文字はアルファベットや数字および一部の記号に限られ、長さは9文字までとなっている。アルファベットは大文字と小文字をそれぞれ自由に使うことができるが、IRC サーバは大文字小文字を区別なく一意に扱い比較するため、例えば BugApple というニックネームを使っているユーザが既にいると、bugapple というニックネームは利用できない。

ニックネームは、IRC を利用している間必要に応じていつでも他とぶつからない限り自由に変更することができる。一方、各ユーザはサーバにつながってるあいだ不変の情報として、ユーザ名、ホスト名、名前を持つ。したがって、時を別にして同じニックネームを使うユーザが複数いた場合、他の人はこれらの情報を元に区別することになる。ただし、プライベートメッセージなどのやりとりにおいては基本的には互いにニックネームを指定して行なう。

IRC サーバ間の接続はさまざまな事情で切れたりつながったりすることもあるが、たまたま二つのサーバ群に分かれている状況下で、それぞれに対して異なるユーザが同じニックネームで接続してしまうことがある。その時点ではそれぞれなんら問題はないが、再び二つのサーバ群がつながり一つの IRC 網になった場合、同じ一つのニックネームを用いるユーザが二人存在することになってしまう。これをニックネームの衝突 (nick collision) といい、これが起こった場合、IRC サーバは両ユーザを強制切断 (kill) することによって回避する。

2.2.2 チャネル

IRC 上で基本となる重要概念としてチャネルがある。IRC はマルチチャネルシステムであり、IRC に参加しているユーザの間でいくつかのグループを作り、そのグループの中だけでそれぞれ会話を行なうようになっている。このグループ単位に会話をおこなう場のことをチャネルと呼び、ユーザは複数のチャネルに属することで同時に複数の会話に参加することができる。

チャネルは誰でも作成することができ、初めてそのチャネルに入ったユーザがチャネルを作成したことになる。一方、みんな抜けてしまって属するユーザがいなくなったチャネルは自動的に消滅する。それぞれのチャネルはチャネル名が与えられて区別されており、各チャネル毎にチャネルのトピックや属性情報やそこに属するユーザのメンバー情報などを IRC サーバが管理している。

IRC ではオープンな場として、話題別、コミュニティ別など、いろんな形で多くの開かれたチャネルが存在する。このような開かれたチャネルでは、同時に多数の相手と自由に会話や議論に参加することができ、リアルタイムにメッセージ交換ができるニュースグループのように使われていることも多い。

また、IRC では誰もが自由に出入りできるチャンネルだけでなく、プライベートなチャンネルを設定することもできる。例えば、チャンネルの存在を隠したり、招待されないと入れない設定にしたり、チャンネルにパスワードをつけたりなど、いろんな意味でプライベートな場を作ることができ、リアルタイムにメッセージ交換ができる内輪のメーリングリストのように使われていることも多い。

このようにチャンネルには種々の属性があり、これをチャンネルのモードと呼ぶ。また、各チャンネルにはトピックと呼ばれる話題などを示すためのタイトルを自由につけることができ、チャンネル内外の者がそれを見ることができる。チャンネルでは各々個別にメンバーシップがサーバによって管理されており、ユーザは明示的に各チャンネルに入ったり抜けたりする。

2.2.3 サーバ

IRC サーバはユーザ同士の待ち合わせ場所であり、talk や phone などと違ってサーバが存在することで各ユーザがそれぞれ非同期にコミュニケーションの場へ出入りして出会うことができる。また、サーバはこれらのユーザ情報やチャンネル情報を維持するとともに、同じ IRC 網の他のサーバへこれらの情報を伝達する。

一方、チャンネル内での会話についてはブロードキャストされず、そのチャンネルに属するユーザが接続しているサーバへだけマルチキャストされる。このため、日本国内のサーバに接続しているユーザしかいないチャンネルの会話は決して国外のサーバへ無駄に伝達されることはなく、また、特に全員が同じサーバを利用している場合はそのサーバの外へは送られることはない。同様に、ユーザ間のプライベートな会話については、それぞれのユーザの接続しているサーバ間でのユニキャストとなる。

現在のバージョンではサーバ同士はループ構造にならないように接続される。その結果、あるユーザから発せられた情報が他へ行き渡る時、そのユーザが接続しているサーバを根とした木構造の中で根から葉へとサーバ上をリレーして順に伝わっていく。そのため特定の個人のメッセージの届く順序は守られるが、それとは別のサーバに接続しているユーザが発したメッセージとの両方を観察すると、IRC 網トポロジーの位置によっては両者のメッセージの順序が逆転して観察されることもありうる。

2.2.4 チャンネルオペレータ

各チャンネルにはチャンネルオペレータという概念があり、そのチャンネルを最初に作成したものがチャンネルオペレータとしてチャンネルの各種属性を自由に設定する権限を持つ。チャンネルオペレータ自体もチャンネルの属性のうちの一つであり、チャンネルオペレータは同じチャンネル内の他のユーザへチャンネルオペレータを分け与えることができる。

2.2.5 チャンネルの各種属性

チャンネルの属性には大きく分けてチャンネルに属しているメンバに付随するものとチャンネル自体の性質との二つがある。どちらもチャンネルオペレータが変更の権限を持つ。

属性の変更のためにはモードフラグとそのパラメータを指定するが、モードフラグ *x* に対して *+x* を指定することでそのモードフラグを設定し、*-x* を指定することでそのモードフラグを解除する。また、パラメータ付きモードフラグの場合は *+x parameter* のように指定する。モードフラグには次のような種類がある。

- p (private)** そのチャンネルに参加していないユーザからチャンネルの存在を隠す。チャンネル名がわかれば外部からも参加しているメンバーなどを知ることができる。
- s (secret)** そのチャンネルに参加していないユーザからチャンネルの存在を隠す。そのチャンネルに入らないと外部からは参加しているメンバーなど一切わからない。
- a (anonymous)** そのチャンネルを匿名モードに設定する。そのチャンネルにおいては誰が発言しても匿名の発言となる。
- n (no message)** そのチャンネルに参加していないユーザが、外部からそのチャンネルに対してメッセージを送ることを禁じる。
- m (moderated)** チャンネルオペレータと *+v* されたユーザ以外のメンバーが、そのチャンネルで発言できなくする。
- i (invite only)** チャンネルオペレータから招待されたユーザと *+I* で指定されたユーザだけがそのチャンネルに入ることができる。
- t (topic limit)** チャンネルオペレータ以外が、そのチャンネルのトピックを変更することを禁じる。
- l (limit)** パラメータとして人数を指定する。そのチャンネルに入ることができるユーザ数を制限する。
- o (operator)** パラメータとしてニックネームを指定する。チャンネルオペレータの権限を与える。
- v (voice ok)** パラメータとしてニックネームを指定する。チャンネルが *+m* の時にも発言できるような権限を与える。
- k (key)** パラメータとしてパスワードを指定する。そのチャンネルに入る時必要となるパスワードを設定する。
- b (bans)** パラメータとして *nick!user@host* のかたちのマスク形式をとり、該当するユーザがそのチャンネルへ参加することを禁じる。
- e (exceptions to bans)** パラメータとして *nick!user@host* のかたちのマスク形式をとり、該当するユーザが *+b* されていても例外として扱われる。

I (invitations) パラメタとして `nick!user@host` のかたちのマスク形式をとり、該当するユーザはチャンネルが `+i` であっても入ることができる。

2.2.6 CTCP と DCC

CTCP とは Client-To-Client Protocol の略であり IRC クライアント間で主に会話以外の種々の情報をやりとりするための共通のプロトコルやメッセージ及びそのしくみを指す。サーバにとっては、CTCP によるメッセージはユーザ同士の単なる会話と区別されず、クライアントがプロトコルを解釈することによって機能する。CTCP にはいくつものコマンドがあるが、実際にはそれぞれのクライアントによってサポートされているコマンドの数も種類も異なる。

CTCP の機能のうちでもっともよく使われるものは、各 CTCP コマンドによる問い合わせに対してその返答を得る機能である。相手に対して各 CTCP コマンドの問い合わせを行なうと、相手のクライアントがそれぞれのコマンドに応じて自動的に情報を返す。チャンネルに対して同様に問い合わせコマンドを投げると、各メンバーが対応していればメンバーすべてから返答を得ることができる。ここでは、ほとんどのクライアントプログラムでサポートしている CTCP のうち主なものを挙げる。

CTCP `clientinfo` 指定したユーザが利用しているクライアントに対して、解釈可能な CTCP コマンドの一覧を問い合わせたり、その返答に用いられる。

CTCP `ping` 指定したユーザに対し、指定した引数をそのまま返すように要求したり、その返答に用いられる。多くの場合は現在の時間などを引数として投げることで、返答が戻って時に相手との間の会話の遅延時間を知ることができる。

CTCP `time` 指定したユーザに対し、相手のローカル時間を問い合わせたり、その返答に用いられる。

CTCP `version` 指定したユーザが利用しているクライアントプログラムのバージョンなどの情報を問い合わせたり、その返答に用いられる。

CTCP `userinfo` 指定したユーザに対し、相手が設定しているユーザ情報を問い合わせたり、その返答に用いられる。

CTCP `dcc` 指定したユーザに対し、指定した DCC リクエストを行なう。

これらの他にユーザは自由に CTCP を定義して実装することも可能なため、例えばおみくじ用の CTCP が作られるなど活用されている例がある。

DCC とは Direct Client Connection/Communication の略であり、IRC クライアント同士が IRC サーバを介さずに直接コネクションを張ることによって、直接会話をしたりファイルを直接送受したり、IRC サーバを通すことによる制限や負荷を避けて直接やりとりをするしくみのことを指す。

一般的にはクライアント同士がなんら情報なくコネクションを張ることはできないが、このための情報を IRC 上にて CTCP を用いて受け渡すことによって相手に伝え、例えばファイルの送受などを IRC をしながら簡単に行なうことができる。

また、DCC を用いて会話を行なうことによって IRC サーバを介さないことで、相手との間にある IRC サーバ間の切断や遅延の影響を受けずに会話を行なえたり、IRC サーバによる送出制限を受けずに直接クライアント同士がファイルを高速に送受することができる。

DCC をするは、最初に一方が相手へ DCC リクエストを行なう。これには、現在のところ、直接会話の開始希望を伝えるを DCC CHAT リクエストと、ファイルの送付要求を伝える DCC SEND リクエストがある。リクエストを受けた相手は、それぞれのリクエストに対して応じたり拒否したりすることができる。DCC リクエストに対して応じると、リクエストをしてきた側へコネクションが張られてそれぞれの DCC が実行される。

2.3 インターネットと IRC の相似関係

IRC の通信はマルチキャストのメタファーでとらえることができる。つまり、インターネットの世界と IRC の世界は相似関係になっており、それぞれで流れる IP パケットと会話のメッセージを対応させると表 2.1 のような関係を見ることができる。

表 2.1: インターネットと IRC の対応

インターネットの世界	IRC の世界
インターネット	IRC 網
ルータ	サーバ
ホスト	クライアント
アドレス	ニックネーム
マルチキャストグループ	チャンネル
ユニキャスト	プライベートメッセージ
マルチキャスト	チャンネルメッセージ

例えばそれぞれの世界での他と区別するための認識 ID はそれぞれ IP アドレスとニックネームであり、IP パケットや会話メッセージを適切な場所へと転送する役目はそれぞれルータと IRC サーバである。マルチキャストグループへの join と leave はちょうどチャンネルへの join と leave に対応しており、ルータやサーバはそれぞれのメンバーがいる最小限の範囲にだけマルチキャストパケットあるいはチャンネルメッセージを届ける。

一方、両者の大きな違いとして 2.2.5 節で示したように IRC ではチャンネルに各種属性がついているため、チャンネル自体が色々な性質を持ったり、チャンネルへの join に対して様々

な制限をすることができたり、チャンネルへメッセージを送るための権限などの細かいコントロールことができるようになっていることが挙げられる。

第 3 章

国内における活動

ここでは、我々が行なってきた国内における IRC の普及活動、日本語の取り扱いに関する諸問題、開発を行なってきたソフトウェアなどについて述べる。

3.1 IRC の普及活動

3.1.1 日本における歴史

日本においても 1989 年終り頃から 1990 年にかけて複数の IRC サーバが立ち上がるとともに接続され、4.2.2 節にて述べる IRC 網を形成していった。具体的には 1989 年 12 月 9 日に fj.comp.comm にて IRC の紹介記事が出たことを経緯に他でもサーバが立ち上がってつなぐるとともにユーザも増えていった。

1990 年 1 月には使用する漢字コードを統一した。これについては 3.2.1 節にて述べる。当時は国内の IRC ユーザ数はかなり少なく、日本における IRC の普及と IRC ユーザ間の交流をはかるため、IRC のユーザ会として 1990 年 3 月に JIRCC を発足させた。これについては 3.1.2 節で述べる。

これらの時期はちょうど WIDE においても東西に NOC が設置されるとともに接続組織の数が増え始めていった頃であるが、当時の国内には WIDE のほかには JAIN と TISN しかなくインターネット利用人口全体が少なかったとともに、IRC の利用者もほとんどが大学および一部の WIDE 接続企業等であった。1990 年のある夜には、IRC 網に接続しているのは全世界で 100 人程度、日本人 30 人程度であり、日本語で話をできるチャンネルが 2,3 あり、全員と知り合えるぐらいの規模であった。このあと、1991 年頃から 1994 年頃までは、大学でのインターネットの開放による利用者の増加により IRC の利用者が増えた。学部学生へ開放した慶應義塾大学の SFC や東京理科大学、早稲田大学等が顕著であった。

現在では日本においても多くの会社や大学などで組織内向けの IRC サーバが立ち上げられているほか、特定の目的のものや汎用のものなどオープンに立ち上がっているサーバや IRC 網も増えつつある。また、1989 年から継続して運用されてきている国内最大の IRC 網は、世界規模の IRC 網である IRCnet の一部を形成しているが、現在は図 4.2 に示すように国内ユーザだけでも常に数千人が接続して利用している。

3.1.2 JIRCC

日本における IRC の普及と IRC ユーザ間の交流ををはかるため、IRC のユーザ会として 1990 年 3 月 31 日に JIRCC (Japan Internet-Relay-Chat Club) を発足させた。これは、当時の IRC ユーザ数が極めて少なく、当初 1 ~ 2 カ月のブームのあとは IRC で待っていても日本人にはなかなか巡り合えないという状況があったため、jircc のメーリングリストの用意、さらには 3.3.2 節で述べる jircc という名のサービス用無人クライアントの開発などによって、IRC の普及をはかるとともに IRC のコミュニティを形成していった。

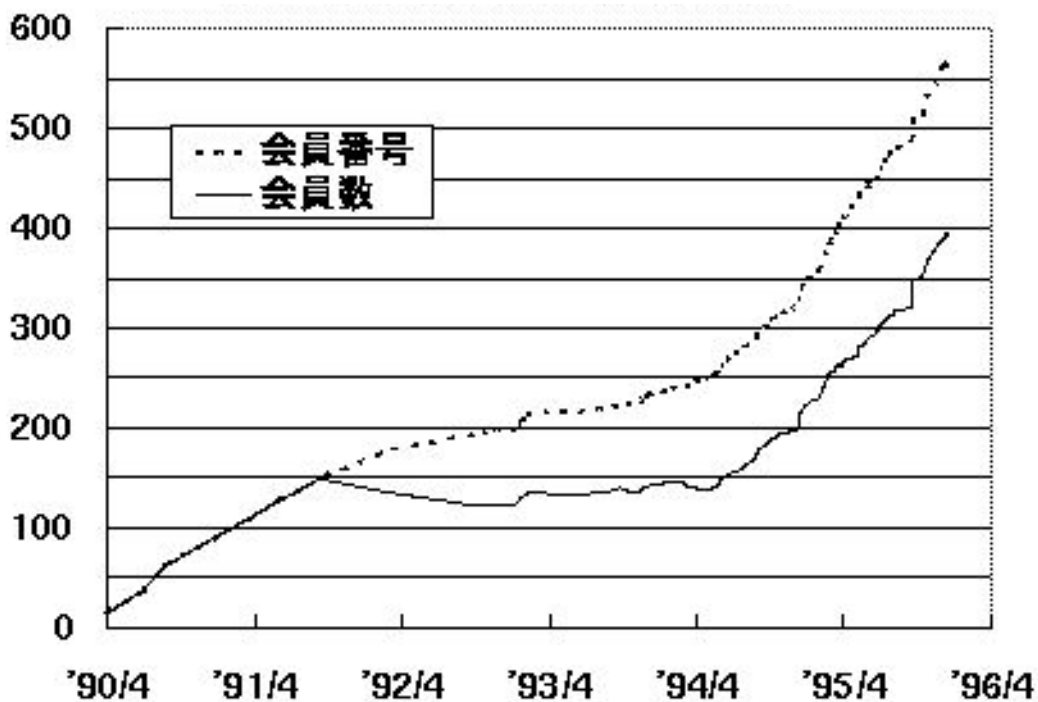


図 3.1: JIRCC における会員数の変化

当時の国内の IRC 人口は数十人程度と IRC 上で全員と知り合える程度の規模であり、国内の IRC ユーザのほとんどは JIRCC へ参加していた。そして、日本語対応化クライアントの情報や国内のサーバ情報などを中心に IRC 全般についての情報交換や議論や告知などが jircc のメーリングリストにて行なわれた。また、当時はまだ WWW が使われていなかったため、空メールを送ると IRC についての情報を集めた文書を自動でメール返送する doc-request サービスを 1991 年 8 月より開始し、のちに <http://irc.kyoto-u.ac.jp/> にて情報提供するようになった。

図 3.1 に示すように 1994 年くらいまでは会員数があまり変わらなかったが、それ以降はちょうど商用 ISP が増えてインターネットが普及してきたため IRC のユーザ数も急激に伸びてきた。最近もサーバの利用者数などを見ると年間 2 ~ 3 倍の伸びを示しており、チャ

ネル名などからみると趣味系のコミュニティが次々と IRC を利用しているのが多いと思われる。これらには、WWW サーバの CGI を利用したチャットの非効率さから IRC へ移ってきたものも多い。

現在は WWW により IRC に関する解説ページが多数出現しているのに加え国内の IRC 人口が数万～数十万いると推測され、普及のために発足した JIRCC の役割は十分果たして終わったと思われるが、現在でも重要な告知や議論を行なうメーリングリストとして機能している。また、IRC に関する発表を中心とする JIRCC 研究会を開催するなどの活動も行なってきた。

また、JIRCC の中に jircc-oper という IRC サーバ管理運用に関する事柄を扱うメーリングリストを設置しており、サーバの設定に関する質問から議論や告知から情報交換まで行なわれてきている。こちらのメーリングリストは jircc メーリングリストの部分集合と位置付けられ、jircc と同様に誰でも参加することができる。

3.2 日本語利用に関する問題

3.2.1 漢字コードの取り扱い

IRC の日本語の会話において用いられる漢字コードが当時の言い方でいわゆる JIS か EUC か SJIS かというのが当初決まっておらず、その頃一般的であった IRC クライアントのソフトウェアが nemacs を用いたものであったために特になにも指定しないと内部コードである EUC を用いて通信していた。

しかし、国際的な IRC 網の環境で用いるということから 1990 年 1 月にメールやネットニュースなどと同様に当時の表現で 7bit JIS コードで統一しようということになり、プログラムに漢字対応パッチをあてて用いることになった。

当時主に用いられていた nemacs は特に何も指定しないといわゆる “ESC \$ @” および “ESC (J” のエスケープシーケンスを発行するようになっていたが、設定によって “ESC \$ B” および “ESC (B” が交じることになり、それらの組合せで 4 種類が入り乱され用いられることになった。

この 4 種類の混在は会話だけで日本語が用いられている間は特に問題にはならなかったが、チャンネル名でも日本語を用いることができるようにサーバソフトウェア開発側と交渉して、1991 年にはいつから漢字を用いたチャンネル名も使えるようになり、チャンネル名の同一性という問題が発生した。つまり、漢字を用いた一見同じに見えるチャンネルが実際は理論上 4 種類存在しうるのである。そして、ほとんどの人は nemacs そのままの “ESC \$ @” および “ESC (J” を用いていたのでその間では支障はなかったが、“ESC \$ B” や “ESC (B” を用いるように特に設定していた人々が漢字を用いた既に存在するチャンネルへ入ろうとすると異なるチャンネルに入って会話ができないということになった。

そこで、漢字を用いたチャンネル名が使われ始めて問題が大きくなり始めた 1991 年 7 月

に、まず第一段階として IRC サーバへのパッチによって当時多数派の “ESC \$ @” および “ESC (J” へと強制変換統一を行ない、最終的には “ESC \$ B” および “ESC (B” を用いることにしたいということから、第二段階として 10 月に後者へと強制変換統一を行なうパッチをあてたサーバへとサーバ側だけ一気に移行を行ない、その間に IRC クライアント側で意識して “ESC \$ B” および “ESC (B” を用いるようにするものを作成し、普及させた。そして、それらが普及し終った 1992 年 5 月に最終段階としてサーバ側での強制変換という特別扱いを中止した。

このような流れのまま現在に至っており、現在でいうところの ISO-2022-JP のさらに強い制限という形になっているが日本語対応の IRC クライアントソフトウェアはこれらの決まりを守ってきている。しかし、1998 年末に登場した DreamCast に載っている IRC クライアント機能は、なぜか行末に “ESC (B” に戻さないという仕様であり、一部問題を起こすことになった。

3.2.2 IRC と ISO-2022-JP

IRC における ISO-2022-JP の利用は、その他にも様々な問題を抱えている。チャンネル名に関する同一性の問題は、“ESC (J” や “ESC (B” の違いによる問題の他に、これらエスケープシーケンスが冗長に用いられた時にも発生する。このように会話の内容などでは問題にならないことでも、互いの使用するものが同一であることを保証するためには唯一となるような正規化された表現が求められる。

次に、EUC と違って ISO-2022-JP のような 0x21-0x7e 部分を用いる文字コード一般に発生する問題がいくつかある。IRC はアルファベット圏で作成されたため、他のプロトコルでも多くの場合そうであるようにアルファベットの大文字と小文字の区別を行わずに同じものと認識する。これはチャンネル名にも適用され、例えば “#ABC” というチャンネルと “#abc” というチャンネルは同じものとなる。これは使っている上でも便利な一面もあるが、ISO-2022-JP を用いる我々は大きな罠に陥ることになる。例えば、同じ理由で “#はる” と “#わに” といった一見異なるものが同じチャンネルとして扱われてしまうのである。これは、それぞれが ISO-2022-JP では “ESC \$ B \$ O \$ k ESC (B” および “ESC \$ B \$ o \$ K ESC (B” となるためである。ちなみに、IRC はフィンランド生まれであることからウムラウト付アルファベット部分である例えば “{” と “[” をも同一視して扱う。

また、IRC ではチャンネル名の区切りに “,” を用いているが、ISO-2022-JP においては “が” や “ハ” などをはじめとして “,” を潜在的に含む文字が非常に多い。これらの文字をチャンネル名に用いるとそこで区切り文字が来たと解釈され、意図したとおりのチャンネル名を作ることができない。

致命的な問題ではないがチャンネルのトピックなどを始めとしてバイト数に制限があるものが色々あり、ISO-2022-JP においては特に ascii と入り交じるとエスケープシーケンスでバイト数を多く使ってしまうために実質的に利用できる文字数が減ってしまうという問題

がある。

これらの他にも現存するほとんどの日本語対応クライアントにおいて問題を起こす可能性があるケースがいくつかあり、そのためにサーバ側でいくつかの対策をとっている。多くのクライアントの実装ではサーバとの通信で送られてきたものをまず一括して内部漢字コードに変換し、その後でプロトコルの解析を行なうが、解析しなければいけない構造の前に漢字が来てしまい、かつ、それがエスケープシーケンスできちんと `ascii` へ戻っていないと後ろの部分が依然として漢字だと思ってしまう解析できなくなるケースがある。完全ではないがこういうケースをなくすためにサーバ側で対策をしている。

また、“A-Z”と“a-z”などのチャンネル名での同一視問題についても、クライアント側で本来はチャンネル名部分であると認識するまでは内部漢字コードに変換しないという原則を守ればクライアントでも同一視できて無用な混乱を引き起こさないが、実際にはほとんどのクライアントの実装ではチャンネル名であると認識できて比較する前に内部漢字コードにしまっているために、同一視をすることができずにユーザに混乱を引き起こす。それを避けるため、実質的に同じチャンネルになるものについてはサーバ側で常に一定のものを返すようにすることでクライアントの実装の問題をカバーしている。これらの日本語問題の対策を行なったサーバを `jp` 版と呼び、国内向けに提供することで国内のほとんど多くのところではこの `jp` 版がサーバとして用いられている。

3.3 開発ソフトウェア

ここでは今までに開発してきた主なもののうち、昔から広く使われてきて他のクライアントへも影響を与えた `irchat` と、世界的に無人サービスロボットの先駆けであり初期の頃に JIRCC の機能を支えるなど普及に貢献した `jircc program` と、IRC に常駐して非同期的利用をする概念を広めるとともにチャンネルの維持やチャンネルオペレータやモードの自動管理などを実現した `pirc` の三つについて簡単に示す。

3.3.1 `irchat`

`irchat` は 1989 年にフィンランドで作成された `elisp` で書かれた IRC クライアントである。`Emacs` が動いていればどこでも使用できること、および、`Emacs` 上では日本語入力機能が容易に使えるため日本語による会話が楽にできることなどから、日本では初期の頃から `irchat` が広く利用されてきた。現在も `nemacs` から `xemacs` まで `emacs` 系のほとんどで動作するように維持するとともに改善しているため、特に `unix` 利用者には多く用いられているとともに `MS-windows` などの環境においても `mule` などを動かして利用しているものも見られる。

日本で実際に使われているのはオリジナルの `irchat` に対して日本語対応及び大幅な機能拡張を行なったものである。ごく初期の頃はオリジナルに対して 3.2.1 節で述べたような

日本語対応を行なっただけであったが、コミュニケーションツールとしてのユーザインタフェースの問題を一挙にいくつか改善するアイデアが出たことを機会にオリジナルとは別れて独自のユーザインタフェースや機能を持つものとして開発を行なっていった。

これらはチャンネル別バッファから始まりワンタッチチャンネル切替やチャンネル順序固定割り当て、圧縮表示機能など多岐に渡り、その後開発されたクライアントの実装にも影響を与えている。特に、チャンネル名の別名機能として“%”で始まるチャンネル名を導入したが、他のクライアントの実装においてもこれに従うものがあり、多くのユーザに影響を与え、IRC の会話中やメールやウェブのページなどでチャンネル名を示す時にまるで“%”で始まるチャンネルが本当にあるかのような誤解のある記述まで見られることが多い。ちなみに、IRC のチャンネル名にはスコープがあり、よく使われている例では“#ちゃんねる:*jp”というチャンネル名は“*.jp”のサーバ間だけで有効となる。これは、トラフィックの観点からも名前空間の点からも可能ならば利用することが勧められるが、irchat にて“%”で始まるチャンネル名に明示的に別名を与えてない場合はデフォルトのマスクを与え、かつ、デフォルトのマスクを明示的に与えてない場合は“:*jp”が初期設定になるので、結果的に、“%ちゃんねる”が標準では“#ちゃんねる:*jp”を示す。会話や文章では、このような対応で“%”を用いることが多くなっている。

3.3.2 jircc bot

IRC に関する情報交換のために JIRCC を発足した当時は初期のブームが去った谷間にあたり、IRC をたまにしか利用しない人に対して IRC 上で JIRCC の発足を知らせることは困難であった。そこで、以下のようなお知らせなど種々の機能持つ無人サービス用の IRC クライアントを開発した。

- 国内初の常駐 bot (常駐型としては海外でも初か?)
- JIRCC への自動勧誘、案内
- JIRCC への入会自動受け付け、会員情報管理
- JIRCC メーリングリストバックナンバー取り寄せ、メンバー一覧表示
- サーバ一覧、サーバオペレーター一覧表示
- NickServ 利用の日本語案内
- IRC 上の掲示板 (news) システム、JIRCC イベント案内
- かな漢字交じり文 separated romaji 自動変換機能

これはニックネームとして jircc を用いており jircc 君と呼ばれていた。また、当時はまだ WWW などがなかったため IRC ユーザ間の情報共有手段として機能した。その頃はインターネット普及が日本より進んでいたアメリカなどからの日本人留学生や日本語専攻学生

などが日本国内からの利用者よりも多く、日本語環境の用意の難しさからローマ字で会話が行なわれる場面も多かったが、そのような状況において会話のかな漢字交じり分を Wnn サーバの逆変換機能および特製辞書などを用いて分かち書きローマ字へと自動変換する機能なども備えていて非常によく利用された。

3.3.3 pirc

pirc は perl で書かれた IRC のツールであり、IRC サーバに対しては IRC クライアントとして振舞い、IRC クライアントに対しては IRC サーバのように振舞う。つまり間にはいつて動作をする代理サーバのようなものであるが、サーバに対してもクライアントに対しても IRC プロトコルを喋るためサーバもクライアントも特別な改造や対応をする必要なく利用することができる。

人がいない間にチャンネルを維持したり会話の記録をするため IRC の黎明期の 1990 年夏より最初の開発を始めた。自分がいない間も pirc がサーバと接続を維持するとともにチャンネルを維持して常駐と会話などの記録を行なう。チャンネルを維持するためのボット用プログラムと大きく異なるところは、自分が IRC する時にはこの pirc に対して IRC クライアントを接続すると pirc はサーバとの接続を無人の時と同様維持したまま pirc を通してサーバを利用することができることである。

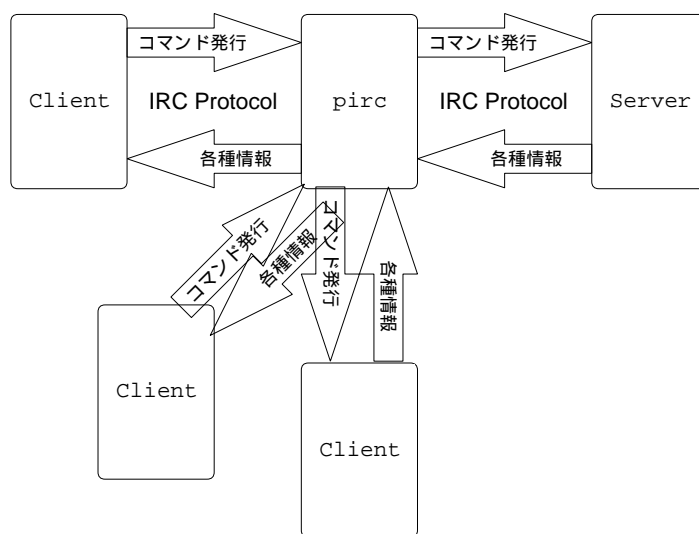


図 3.2: pirc へ複数の IRC クライアントを接続

一般に、図 3.2 に示すように pirc に対して IRC クライアントを 0 個から任意の個数接続することができ、複数の IRC クライアントを接続した場合はどちらからも操作することが可能である。例えば職場で X ウィンドウシステムと X 上の IRC クライアントを動かして

繋げっぱなしであるところに、家にかえってモデムから入って irchat をさらに繋げて使うといったようなこともできる。また、場所を移動して端末が変わる時にそれまで使っていた irchat を pirc から切り離して終了して、移動先の端末の MS-Windows 上で異なる IRC クライアントを動かして pirc 繋げ直して会話を再開、といったこともできる。このあいだ、pirc は IRC サーバへ接続をずっと維持しているためチャンネルから抜けることもなく、行なわれている会話を落とすことはない。

典型的な使い方としては、ダイアルアップユーザがどこかで確保したインターネットへの常時接続ホスト上で pirc を動かしておき、ダイアルアップでインターネットへつなぐたびに手元の IRC クライアントをその pirc へつなぐというやりかたである。この時、pirc は IRC クライアントから接続を受けるとそれまでに pirc がサーバから受けたメッセージの一定量のバックログを IRC クライアントへ送付し、ダイアルアップでつないでいなかった間の分の会話を見ることができる。また、pirc は会話などの情報をファイルに自動的に記録するため、自分がいない間の溜った分が多くてもまとめてみることもできるし、いた間の分も含めてあとで検索をしたりする利用も考えられる。

このように、自分がいない間の会話を見てあとで反応することも可能なため、非同期にもインタラクティブにも IRC をすることを可能にするための支援ツールとなっており、pirc を用いることで IRC の利用方法の幅が大きく広がった。

このほか pirc 自体の利用法としては、IRC の代理サーバの性格より社内など直接インターネットに出られないユーザがゲートウェイ上で pirc を動かすといった利用も行なわれている。なお、IRC クライアントの pirc への接続はパスワードなどで制限されるが、これは IRC サーバの運用でパスワードを付けて動かす場合とおなじしくみで実装している。

IRC クライアントを接続するかしないかに関わらず有用な pirc 自体の機能としては、IRC サーバとの間がなんらかの原因で切れてしまった時に必要ならば再試行して接続を復活したり、そのときに入っていたチャンネルにすべて自動で入り直したり、もしもニックネームが他の人に使われていたり一時的に利用できない場合は復活が可能かどうかを自動的に再試行し続けたり、チャンネルから蹴り出された場合に可能ならば再び戻ることを試みたり、といった数多くの機能が挙げられる。

また、自分が指定した特定の人たちに対して自動的にチャンネルオペレータの権限を配布したり、チャンネルのモードを指定したように維持したりといったことも各機能毎にモジュールが用意されていて使いたいものを選んで利用できる。

これらのモジュールは pirc の拡張用フック機能を利用して作られており、会話の中の指定した特定のメッセージに反応して自動的に返事をするようなボット機能も簡単に作れるようになっている。例えば、チャンネルに常駐をしない人向けに伝言を預ったりする伝言ボット機能もこのような形でメッセージに対するフックとして実装されており、伝言ボットモジュールとして提供されている。

現在も pirc は IPv6 サーバへの接続対応といった感じで開発が進んでいる。一方、このような特徴と概念を備えた pirc と同じ部類のものとして plum というものが最近では作られ

ていたり、pirc から分家した madoka が機能拡張を進めている。これら全体として利用者数はますます増えており、これからもこのような形態で IRC を利用する人々は大きく増えていくと予想される。

第 4 章

サーバ運用の状況

4.1 サーバ運用管理

IRC サーバは、サーバ自体が他のコマンドを実行したり関係のないファイルにアクセスしたりすることがないので、他のサーバよりもセキュリティは比較的安全である。サーバを動かすための専用のユーザを作って安全度を増したり、設定によっては chroot をするようにもできる。我々が管理している IRC サーバのホストではさらに安全を確保するために一つ余分に IP アドレスを割り当てて IRC サーバ専用に使っている。そして sshd などは本来の IP アドレスのほうで動かし、完全に分離している。

サーバ間接続については、相手の情報とパスワードを自分のサーバに設定し、自分の情報とパスワードを相手のサーバに設定してはじめて互いに接続することができる。このとき IRC 網のなかで自分と相手のどちらが上流か下流かを可能な限りはっきりさせておくほうが好ましい。そして、自動的に相手へ接続をしにいく設定は必ず下流側から上流側へつなぎにいくようにするべきである。これを守らないと、互いに同時に接続しにいて失敗したり、あるいはいつまでも他と接続を持てない状況に陥る。

4.2 IRCnet

ここでは、WIDE プロジェクトのサーバが接続参加している国際的な IRC 網である IRCnet に関して、その状況と運用について述べる。

4.2.1 国際的な状況

インターネット上には数多くの IRC 網があり、それぞれの参加サーバ数やユーザ数などの規模も様々である。すべてのメーリングリストを把握するのが困難であると同様にすべての IRC 網を把握するのは当然困難であるが、<http://www.irchelp.org/irchelp/networks/> において以前は IRC 網の情報登録を受け付けていたために現在 99 個の IRC 網についてのまとまった情報を見ることができる。

このように多くの IRC 網がある中で、常に数万規模の接続クライアント数を保つ EFnet, Undernet, IRCnet, DALnet の四つは特に Big Four と呼ばれている。他の三つがどれも参加国数が十程度なのに対して IRCnet は現在三十以上の国々で上がっているサーバ群を接続している最も国際的な IRC 網である。

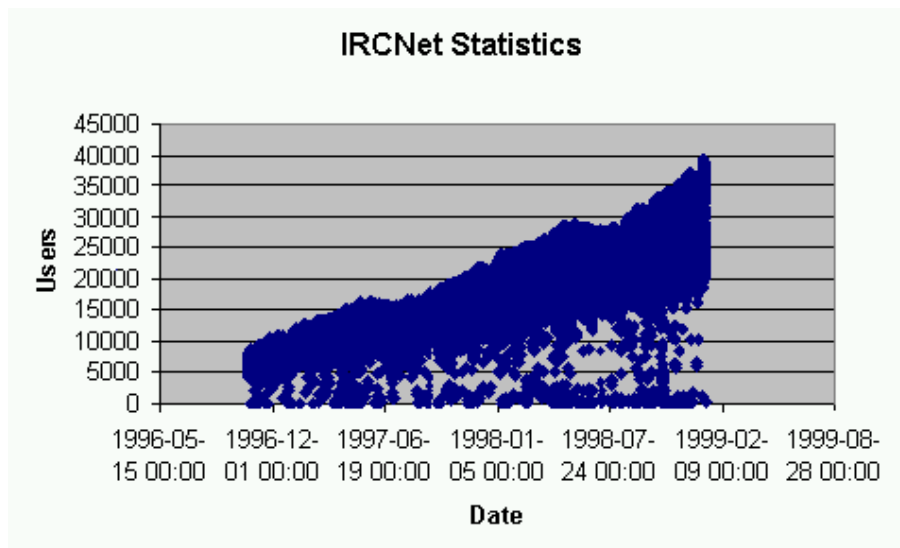


図 4.1: IRCnet におけるクライアント数の変化

IRCnet は IRC が作られた 1988 年から続いている IRC 網である。当初は国際的な大きな IRC 網は唯一であり、その成長過程で IRC 網の構成要素である各サーバは新規接続されるものもあれば廃止されるものもあるなど入れ替わりは多く発生している。また、種々の理由により集団離脱や分裂することもあり、大きな分裂としては 1996 年 7 月に規模の問題や運用ポリシーの違いなどから北米中心のサーバで構成される EFnet と大きく二つに分離した。

現在 IRCnet のサーバが上がっているところは 30 カ国、運用に参加しているのは 46 カ国であり、AU, NZ, BE, DE, DK, IS, ES, FI, FR, GR, AL, BG, CY, MK, TR, YU, IL, IT, JP, MY, NL, PL, RU, BY, KZ, SU, UA, UZ, TJ, AM, KG, TU, AZ, MD, GE, SE, SK, CZ, HU, RO, AT, SI, UK, IE, US, CA といった国々で構成されている。

このように多くの国々のサーバ群が接続しあって一つの IRC 網を形成しているが、各サーバを運用している組織も様々であり、商用 ISP をはじめとして企業や大学から公的なところや WIDE のようなところまで種々雑多である。このような中で明示的な厳しい規則を設けることなくオープンなメーリングリストを中心に協調的な運用が行なわれている。同時にサーバオペレータだけのメーリングリストもあって使い分けられている。

接続にあたって守るべきことや運用にあたって守るべきことなどについて細かい規定が設けられていない分、なにかまずいことが発生すれば IRC 上で一気に判断して決めて特定

のサーバや国全体を一時的に接続切断したり、時間が経っても問題が解決されない場合には永久追放をとることもある。しかし、意見が別れて決定方法などが明確になっていないことが問題になる場合もあり、議決規則などを設定する動きも進められている。

4.2.2 国内での状況

国内では 1989 年末頃から複数のサーバが立ち上がるともに順次接続されていったが、日本で最初の `utsun.s.u-tokyo.ac.jp` が現在の IRCnet につながる IRC 網に US のサーバを通して接続していたために、必然的に国内のサーバ群は当時唯一であった国際的 IRC 網を形成するメンバーとなった。

この 10 年間にこの IRC 網にサーバを接続した組織は、`aist-nara`, `astem`, `dti`, `hokudai`, `jaist`, `kaba`, `karrn`, `keio`, `kyoto-u`, `kyushu-u`, `kyutech`, `mei`, `nagoya-u`, `omron`, `osaka-u`, `sony`, `tdi`, `titech`, `tohoku`, `tokai-ic`, `ube-c`, `u-tokyo`, `waseda`, `wide`, `yamaguchi-u` など数多く様々であった。そしてそれらの組織から 40 以上のサーバが接続される一方、多くのサーバは種々の事情により廃止されたり接続切断された。その理由は、管理者がその組織を去るなどの人の移動やマシンやネットワークが負荷に絶えられなくなった場合など様々である。このように新陳代謝を繰り返しながらも全体としては継続的に運用されてきた。それとともに安定したサーバが生き残ったり IRC 網に収容可能なクライアント数が増えたりと、少しずつではあるが確実に状況の改善が進んでいる。

表 4.1: IRCnet の国内のサーバ

サーバ名	ポート
<code>akiu.gw.tohoku.ac.jp</code>	6667
<code>irc.dti.ne.jp</code>	6666 - 6667
<code>irc.huie.hokudai.ac.jp</code>	6667
<code>irc.karrn.ad.jp</code>	6660 - 6669
<code>irc.kyoto.wide.ad.jp</code>	6660 - 6669
<code>irc.rcac.tdi.co.jp</code>	6660 - 6669
<code>irc.tokyo.wide.ad.jp</code>	6660 - 6669
<code>irc6.rcac.tdi.co.jp</code>	6667 (IPv6)
<code>race-server.race.u-tokyo.ac.jp</code>	6667

マシンのパワー増大などによりクライアントの収容接続可能数が大きなサーバが増えたことから、ここ数年間は国内は 10 個程度のサーバ数で構成されている。現時点でのサーバ設定およびマシン能力ではこれら全体で少なくとも約二万ユーザは同時収容可能であり、さらにマシン能力向上を含む今までのサーバの移設や復活などによる大容量サーバの増設

計画も進められている。

また、昨年からは IPv6 環境の IRC サーバを `irc6.rcac.tdi.co.jp` で立ち上げて実験運用を行っており、WIDE 6Bone を通して世界中から IPv6 にて IRCnet を利用することができるようになっている。

4.2.3 サーバのユーザ受け入れに関する現行ルール

IRC サーバは受け入れ可能なユーザ集合を指定することができ、各サーバ毎に受け入れるユーザのホスト名や IP アドレスの範囲を設定することで指定される。

サーバやユーザの数が少なかった頃はほとんどのサーバにおいて任意のホストからのユーザ接続を受け入れていたが、世界的に複数のサーバが分布するなどサーバの数や接続ユーザ数が増えるとなると、ユーザを適切な比較的最寄りのサーバに誘導すること、および、各サーバにおける受け入れユーザの管理の手間の負荷分散という観点から、現在は以下のようなユーザ受け入れ担当ルールで運用されている。

サーバを立ち上げている TLD のところでは、自分の TLD 内のユーザをその TLD 内のサーバ群内において受け入れる。つまり、TLD 内のユーザだけを受け入れるという設定をしてもよいが、TLD 内のユーザは TLD 内のサーバ群内において収容できるようにする。

厳密には、BODY とよばれる管理単位の集合で IRCnet が構成されており、多くのユーザを抱える TLD は単独で一つの BODY を、少ないところではいくつかの近隣諸国同士が複数の TLD で一つの BODY を構成している。そして、大半のサーバは自分の TLD に加えていくつかのドメインをカバーするという自 TLD 内オープンサーバとして運用されている。サーバ数の少ないところでは近隣諸国で互いにカバーをしている例も多い。サーバがまだ立ち上がっていないところのユーザ収容問題などもあるため、各地にいくつかのグローバルオープンサーバや、サーバが立ち上がっている TLD などの一部受け入れを除いた準グローバルオープンサーバが立ち上がっていて世界中のユーザをカバーをしている。

このような状況の中、jp TLD においては単独で一つの BODY を形成しており、各サーバにおいて `*.jp +` という形で主に国内のユーザを受け入れている。ただし、各サーバによって `+` の部分は異なるが、どのサーバも少なくとも `*.jp` のユーザを IRCnet に接続収容できるようにカバーして運用されている。また、各サーバにおいて自分の組織や特定の優先させたい組織のユーザの受け入れ枠を確保したいという要望とのバランスから、次のようなルール (通称 50% rule) で現在運用が行なわれている。

各サーバの受け入れ可能なユーザ数のうち、その半数までは自由に自分の組織や特定の優先させたい組織のユーザ用に確保して運用してもよい。残り半数以上については `*.jp` のユーザを受け入れて収容できるようにする。

IRCnet-jp へのサーバ接続条件としては、以上のユーザ受け入れに関するルールの他にも、サーバの受け入れ能力、マシンや環境の安定度、ネットワークの質、管理者の IRC に関する技術レベル、実際のサーバ運用管理においての多くの必須事項など数多くの条件を満たす必要があり、接続後も協調的運用が求められる。このため、いままで 9 年間には 30 近いサーバが条件を満たせずに廃止や移動したり強制切断させられたりして絶えず新陳代謝を行っており、IRC 網全体としてはレベルアップしながら継続的に長期運用が実現されてきている。

なお、国際的にも明文化されていないルールや各サーバへの改善要請などがいくつもあり、それらに対処したり改善する形で IRCnet-jp も運用されてきている。また、これらのルールの明文化や各種決定などについてもっと実効力をもたすため、IRCnet の国際レベルにおいては投票制度までも含めた協調運用の改革について話し合いが進められている。これらの国際的な取り決めや各種要請については今までと同様に IRCnet-jp におけるローカルルールよりも優先される。

また、これらの制約により他の IRC 網と相互接続という形をとることはない。IRCnet へのサーバ収容接続条件 (国際および日本ローカル) などに照らし合わせて各サーバ個別に許可できるかどうか判断する。そのため、必ずしも全てのサーバが OK にならない可能性もある。一方、接続トポロジについてはいろんな意味で効率的になるように柔軟に配慮が行なわれる。結果的に、新たに IRCnet に収容される各サーバが元々接続していたトポロジそのままに接続されることもありうる。

4.2.4 IRCnet の運用管理における新たなモデルとルール作り

IRCnet の運用管理のルールのうち 4.2.3 節に書かれている IRC サーバのユーザ受け入れに関するルールに対して、IRCnet へ接続を希望する商用 ISP から新たな要求がでてきた。それらは、ユーザとして自分の顧客のみ収容したい、責任ある立場でユーザを収容したい、といったものであり、そのままでは現行ルールとうまく合わないためになんらかの解決案を考えることになった。そして、IRCnet の運用管理における新たなモデルとルール作りの議論が現在進められている。Internet Week'98 においてこれを議題に BOF を開催し、その時点までに議論が行なわれて提案されているものとして以下のような叩き台を示すとともに議論を行なった。

1. 日本 IRC 次世代運用検討会 (JIRCC-ng) の趣旨

現在、国内 IRC 網 (IRCnet) は、WIDE や一部の大学、企業、ネットワーク組織、ISP 等によって運用されているが、IRC 利用者の大半は ISP 経由で IRC サーバに接続しているにもかかわらず ISP にはサーバはほとんど存在しないということからサービスの配置的、あるいは計算機資源やネットワークトラフィ

ックのコストのアンバランスが問題視されるようになってきた。また abuse の増加により管理面のコストも増大してきている。

そこで、国内網の全体コーディネーション、技術レベルの維持、abuse 対策等の観点から国内における IRC 網の次世代の運用管理体制への模索が開始された。本検討会では、国内 IRC 網の円滑な運用管理方法を検討しそれらを実現するための新たなモデルとルールを構築することを目的とする。

また、複数の ISP が網に参加することによって発生するであろう商用 ISP 間の相互接続に関する様々な課題の解決も同時に検討課題としている。

2. 国内 IRC 網の新しい運用管理に対する要請

IRC 網は網全体の安定性を考慮した場合、無秩序に接続サーバを増やす発展形態をとり得ない。従ってサーバの接続には何らかの制約を設けざるを得ない。現在の IRC 網でのサーバ接続の基準はおおよそ、

- (1) IRC Server をきちんと管理運営できる事 (abuse 等の対応含む)
- (2) *.jp を 50% 收容し、それ以外のドメインのユーザーも 50% 收容する事
- (3) IRC Server 1 台に関して 2000 名程度のユーザーを收容できる事

等を目安にしてきた。しかし、大規模化に伴うコストの分散や一部で商用 ISP 間のサーバ接続が発生することを考えた場合、特に (2) の接続条件について管理するユーザの範囲や abuse 対策の対象となる 範囲が網の規模とともに拡大してしまいスケールしないことが問題となることが明らかになった。

また、IRCnet の国際接続に関しては、現状、

- (a) 国単位で管理を分割してその中でユーザを收容する

という基本方針がとられている。そこでこの基本方針を踏襲した上で、国内でも管理範囲を分割できるようにする方向で接続条件を検討した結果、国内 IRC 網の新しい管理運用体制には、

- (A) IRC Server をきちんと管理運営できる事 (abuse 対応, 安定接続の維持)
- (B) 国内 IRC 網を木構造で関連づけられた管理主体の元に運営する事
- (C) Server Operator はこれまでと同様 ML 等で確実に情報交換を行う事
- (D) 各管理主体が国内のユーザーをできるだけ洩れの無い形で收容する事

等が要請されるとの結論に至った。

3. 運用管理体制のモデル（素案）

2. で述べた条件に沿って新たな運用管理体制に参加する各管理主体が持つであろう木構造の階層をその管理主体に属するサーバの役割によって大まかに分類すると、

- 第一階層 国際 Gateway HUB サーバを持つ管理主体のグループ (*.jp)
- 第二階層 上位 HUB サーバを持つ管理主体のグループ (Hub of Hub)
- 第三階層 下位 HUB サーバを持つ管理主体のグループ (ISP 等)
- 第四階層 リーフサーバのグループ

のように分けられる（複数の階層に同時に属するものも存在し得る）。このうち、第一階層から第三階層までのサーバの運用管理には該当サーバ間の調整以外に上流の管理主体との調整が必要になるため、これらを管理階層の範囲とみなすことができる。このモデルでは下位の管理階層の接続コーディネーションを上位の管理階層で処理することで管理が分散されるが、こうすることで必然的に下位サーバの接続条件が上流の管理主体から示される形になり、網全体の不安定化につながる無秩序な網の拡大の抑制が期待できる。

このモデルでは各管理階層に属するさまざまな管理主体が考えられるが、管理主体の最小単位は、暫定的に「JPNIC 会員でありかつ AS を持っている主体」とすることが提案されている。もちろん、いくつかの管理主体が集まって複合管理主体を形成しても構わない。

以上が、日本 IRC 次世代運用検討会 (JIRCC-ng) において検討されている、国内 IRC 網の新たな運用管理体制の素案である。

4. 国内 IRC 網の新たな運用管理体制についての議論の情報共有

JIRCC-ng での議論に際し、情報を共有するためにメーリングリスト (irc-ng@irc.kyoto-u.ac.jp) が開設されている。このメーリングリストは運用の枠組が出来次第解消するものとして最初は JIRCC-ng のメンバーが参加するものとする。

5. 付録

実運用では、今後

- 従来通り 50%ルールに基づいて*.jp の範囲でユーザを收容し、abuse 対策は K-line でやっていく
- abuse 対策とさらに下流での管理の分割のために收容ユーザの範囲を管理主体が接続性を管理している顧客に限る

という2つのポリシーに基づく管理主体のグループが共存する形になるが、前述の要請を満たしている限り、新たなポリシーのグループの参加は妨げないものとする。

4.2.5 接続クライアント数の変化

図 4.2は 1999 年 4 月 26 日午前 3 時半時点の、IRCnet の国内サーバそれぞれに接続しているクライアントの数を合計したものの一日の変化のグラフである。これは、その晩のピークは約 7500 であり昨晚のピークは約 8700 であったことを表している。ちょうどそれぞれ日曜の深夜と土曜の深夜に該当する。

図 4.3と図 4.4と図 4.5は、同様の時間帯のグラフであり、それぞれのサーバに接続しているクライアントの数の一日の変化を表している。これを見ると、それぞれのクライアント数こそ違え、グラフの形はすべてほぼ相似となっている。それぞれのサーバには異なるユーザが接続して利用しているが、このようにサーバが異なってもグラフの傾向がほぼ同じになるというのは IRC ユーザ全体の平均的行動を示しているものととらえることができる。どのサーバにおいても 23 時頃から急激にクライアント数が増え、深夜 1 時頃をピークとしてその後徐々に減っていき、朝 8 時にストンと落ちて、その後は急激な変化はなくならである。

図 4.6と図 4.7と図 4.8と図 4.9は、irc.tokyo.wide.ad.jp におけるクライアント数のそれぞれ日・週・月・年の変化である。これらを見ると、一日の変化の傾向は他の曜日でもほぼ同じであることがわかる。8 月に現在利用している PC へと移設をしてからは、順調に接続クライアント数も増えていき、4 カ月前と比べるとでほぼ倍増していることがわかる。

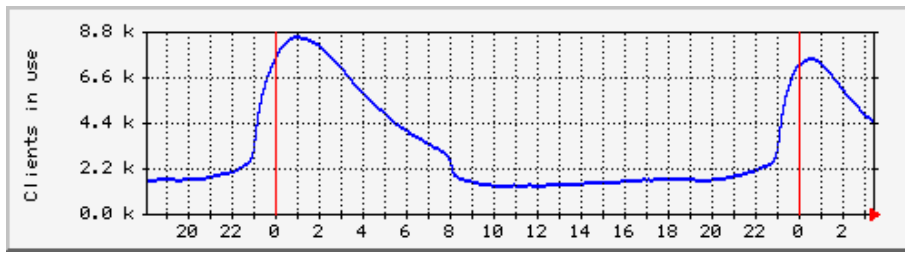


図 4.2: IRCnet 国内サーバにおけるクライアント総数の一日の変化

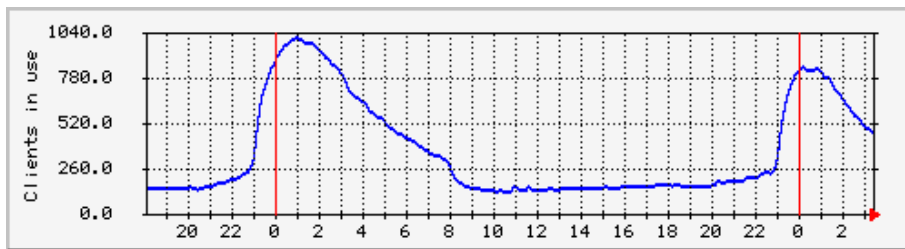


図 4.3: irc.karrn.ad.jp におけるクライアント数の一日の変化

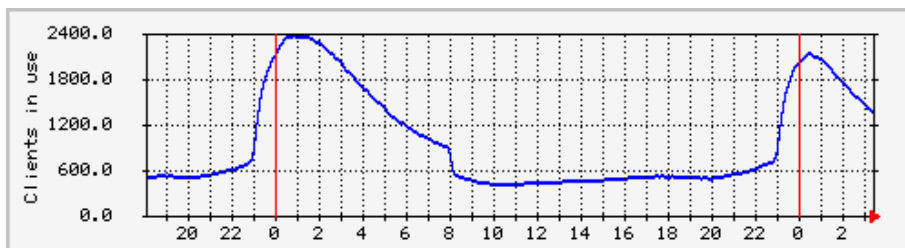


図 4.4: irc.rcac.tdi.co.jp におけるクライアント数の一日の変化

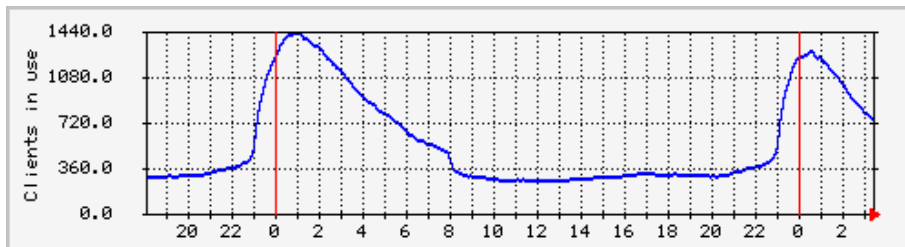


図 4.5: irc.kyoto.wide.ad.jp におけるクライアント数の一日の変化

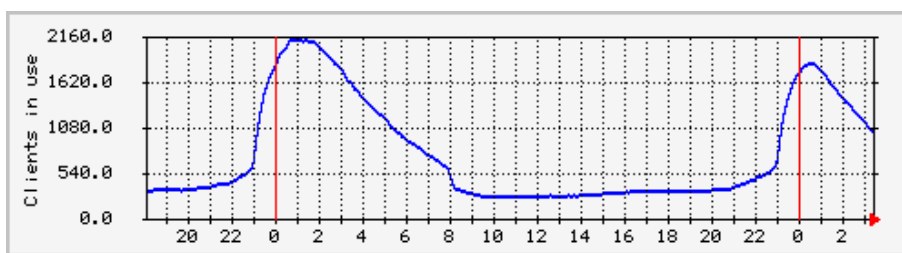


図 4.6: irc.tokyo.wide.ad.jp におけるクライアント数の一日の変化

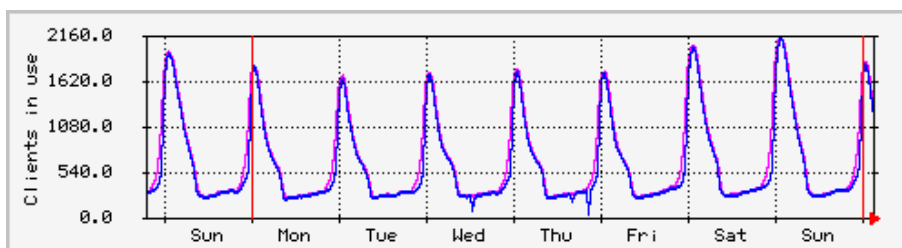


図 4.7: irc.tokyo.wide.ad.jp におけるクライアント数の一週間の変化

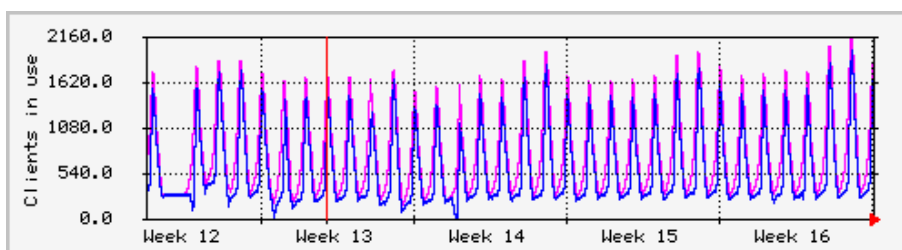


図 4.8: irc.tokyo.wide.ad.jp におけるクライアント数の一カ月の変化

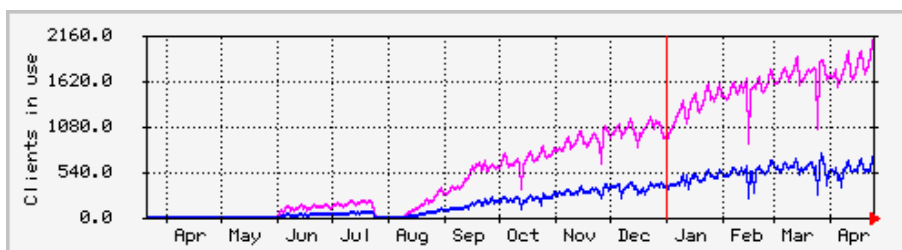


図 4.9: irc.tokyo.wide.ad.jp におけるクライアント数の一年間の変化