

第17部

IRCの運用技術と活用技術

第1章 はじめに

IRC ワーキンググループは、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い、それら技術の開発と確立を目標とする研究グループである。

IRC は 1988 年にフィンランドで開発されてから以来、その利用と運用も含め様々な問題に対して対応するために多くの拡張と改良が加え続けられてきている。

また、以前からこのワーキンググループのうちの多くのメンバーによって IRC に関する様々な活動や開発を行なってきたり、日本において IRC の黎明期より IRC の普及と発展を支えてきている。

今回の報告書では、以下の内容に関する報告を行なう。

第2章では、これまで irc.tokyo.wide.ad.jp や irc.rcac.tdi.co.jp などにおいて運用保守の利便性やセキュリティに重点をおいて構築してきた IRC サーバの動作環境についてのノウハウを、今回新たに設置することになった irc.fujisawa.wide.ad.jp に持ってきてゼロから構築することで整理をし、それらの手法と技術をまとめて報告する。

第3章では、IRC サーバ実装の現状と日本語を用いる場合の問題と解決について述べる。

第5章では、WIDE の IRC サーバが参加している IRC 網である IRCnet の現状と問題点について述べる。

第4章では、我々が運用している各サーバにおいて計測したデータを用いて IRC の利用状況の統計をまとめ、これらを分析して報告する。

第2章 IRC サーバ動作環境の構築

2.1 藤沢 NOC における IRC サーバの開設

WIDE プロジェクト藤沢 NOC では、2000 年 3 月から IRC サーバの構築を始めた。藤沢 NOC では、過去に endo.wide.ad.jp が IRCnet に接続し運用されていたが、マシンの負荷問題や管理者不在などの理由から廃止されていた。しかし、近年 IRC のユー

ザ数は飛躍的に増加しており、IRC を用いたコミュニケーションは日常、研究活動において不可欠な要素となっている。このような IRC 網への接続サービスの重要性を背景に、再び藤沢 NOC に IRC サーバ (irc.fujisawa.wide.ad.jp、以下 irc.fujisawa と呼ぶ) を開設した。

現在、irc.fujisawa は SFC の村井研究室内部でのコミュニケーションに使用している。それと共に、IRCnet への接続、運用準備を進めている。IRC サーバは様々な攻撃の標的になりやすいため、セキュリティの強化や、悪意を持ったユーザからのサービスの妨害を予防する十分な準備が必要である。

2.2 マシン環境

irc.fujisawa は、現在 Sparc Station20 で構築している。CPU は TMS390Z50(50MHz)、メモリは 96MBytes である。オペレーションシステムには Solaris2.7 を用いている。現状ではサーバのスペックに不安があるため、十分なパフォーマンスが得られない場合、メモリの増設やマシン自体の交換などを考えている。(例えば、最大で 4000 以上のクライアントが接続する irc.tokyo.wide.ad.jp では、128M のメモリのうち半分が ircd によって使用されている。)

2.3 ネットワーク接続

irc.fujisawa は 10Mbps の Ethernet により藤沢 NOC に接続している。irc.fujisawa では物理インタフェースに設定した IP アドレスとは別に、ソフトウェアインタフェースを用いて IRC サーバ専用の IP アドレスを設定している。ソフトウェアインタフェースは物理インタフェースの alias によって実現した。

IRC サーバ専用の IP アドレスは、物理ネットワークの IP アドレス空間とは別のアドレス空間から取得している。この IP アドレスへの経路は、irc.fujisawa 自身が WIDE の OSPF に参加しホストルートとして伝搬させている。

このように、サービスに用いる IP アドレスを別に持つことは次の利点がある。

- サービスを他のホストに移設しやすい

サービスホストに問題が生じた場合、ソフトウェアインタフェースごと他のホストへ移設

すると、移設後にオペレーションの手間がかからない。名前空間を変更する必要がないため、即座に代替サーバにサービスを移行できる。また、IRC のようにサーバ同士が接続している場合、接続先のサーバの設定ファイルを変更する必要がない。

- IP アドレスの逆引きをそのサービスホスト名自身に設定できる

サービスホストとしての名前を、実際のホスト名とは別に設定できる。このため、IRC サーバのようにサービスホスト自体から他のサービスホストへコネクションを張るようなタイプのサービスでは、相手側におけるホスト名認証でサービスホスト名を用いることができる。

- セキュリティの強化

サービスホストは侵入などの攻撃を受けやすい。そのため、ネットワークからのアクセスを厳しく制限する必要がある。しかし、外部からのアクセスが全くできない状態では管理が困難である。管理用のアクセス先 IP アドレスとサービス用の IP アドレスを分け、サービス用 IP アドレスにより厳重な制限を設けることで、管理者の作業を阻害せずにサービスホストのセキュリティを高めることができる。

- サービスのバックアップ体制がとれる

応用例として、例えばサービスホストがダウンした場合、経路制御によって自動的にバックアップのサービスホストへ到達できるシステムが考えられる。これは、同時に複数のホストにおいて同じ IP アドレスのソフトウェアインタフェースを設定し、それぞれからホストルートを伝搬させ、それらの経路アナウンスメトリックに差をつけることで実現できる。irc.fujisawa では、まだこのような使用法はとっていないが、将来的に有効なバックアップ方法である。

2.4 セキュリティ対策

2.4.1 アクセス制限

管理者が ssh でマシンにアクセスする場合、次の

ような制限を設けている。

- 物理インタフェースの IP アドレスを用いる。IRC サーバ専用の IP アドレスに対するアクセスは許可しない。
- 指定された IP アドレスからのアクセスだけを許可する。
- Unix Password による認証を許可しない。RSA 認証を用いる。
- root アカウントでのアクセスを許可しない。

また、telnet や finger など ssh 以外のポートへのアクセスは、IRC サーバを除いて許可されていない。

2.4.2 プロセスの制限

irc.fujisawa では、irc 関係のプログラムを実行するために専用のユーザ ID を設けている。これらのユーザには/etc/passwd において login shell が設定されておらず、通常のユーザとして login することはできない。

- irc(ircd,exec-ircd)

exec-ircd は ircd を自動的に起動させるデーモンプロセスである。これについては後述する。exec-ircd の実行ファイルは、起動されるとプロセスのユーザ ID、グループ ID をを irc へ切り替える。ircd は exec-ircd から execute されるため、ユーザ irc のプロセスとして起動される。

- ircstat(クライアント数の監視)

ユーザ ircstat の crontab を作成し、1 時間に一度 ircd が使用しているコネクションの数を記録している。

- ircwatch(pirc)

pirc は IRC サーバとクライアントの間に入り、代理サーバに近い振舞いをするツールである。IRC サーバに対しては IRC クライアントとして接続し、IRC クライアントに対しては IRC サーバのように振舞う。pirc は IRC 黎明期の 1990 年から開発され、人がいない間にチャンネルを維持したり会話を記録するために使用された。irc.fujisawa で

は、これを利用して&ERRORS、&LOCAL、&SERVERS、&KILLS など特殊チャンネルのログを記録している。また、pirc は拡張用フック機能を持っており、特定のメッセージに反応して追加モジュールを呼び出すことができる。これを利用して特殊チャンネルの重要事項を拾い、指定したオペレータへの通知を行なうモジュールが実装されている。藤沢の管理者は常に irc.fujisawa にクライアントとして接続しており、この通知を IRC のプライベートメッセージとして受けとる。pirc は、ユーザ ircwatch の crontab を利用して常に起動されている状態に保っている。

また、もしも ircd がクラックされた場合にも、侵入者が他のファイルを操作できないようにするため、次の措置をとっている。

- ファイルシステムのルートパスを変更

irc.fujisawa では、ircd を実行するために必要な最小限の環境を /home/irc/root に集めている。これを ircd のルートパスに設定し、ircd がアクセスできるディレクトリを制限している。実際には、ircd の起動を exec-ircd デモンが行なう。exec-ircd は起動されると /home/irc/root を自らのルートパスに設定する。ircd がダウンしている場合、exec-ircd は fork し ircd を execute する。ircd 自体も chroot の機能を持っているが、ircd が chroot するのを待つよりも、すでに chroot したプロセスから ircd を execute したほうが安全である。

- グループ制限により、他のファイルを不正に書き換えられないようにする

上で述べた専用ユーザが属するグループをグループ irc とし、ircd の設定ファイルを操作する管理者をグループ ircadmin に設定している。/home/irc 以下の全てのファイルは、owner グループを ircadmin に設定し、ircd などのプロセスがこれらのファイルに不正にアクセスできないよう制限している。唯一 /home/irc/root/var は、pid ファイルや log ファイル作成のため owner グループを irc に設定し、ircd からの書き込みを許可している。

2.5 Solaris2.7 での特殊な設定

2.5.1 ライブラリ、デバイスファイルの複製

2.4.2 で述べた通り、ircd は chroot によってルートパスを /home/irc/root に設定しているため、共有ライブラリを参照できない。そのため、ircd をコンパイルする際に静的リンクが必要になる。しかし、Solaris2.7 では静的リンクがサポートされておらず、これを実行しようとした場合、dlopen などの関数についてライブラリが用意されていないことが分かった (共有ライブラリのみ)。そのため、動的リンクのまま ircd をコンパイルし、必要な共有ライブラリを /home/irc/root/usr/lib 以下に複製した。また、malloc() ルーチンは疑似デバイス /dev/zero にアクセスし、0 に初期化されたメモリ領域を獲得する。このため、これを /home/irc/root/dev/zero に複製した。

2.5.2 プロセスが開けるファイルディスクリプタの上限を変更

ircd は、他の IRC サーバやクライアントなど非常に他数の通信相手を持つ。irc.fujisawa では、ircd の最大コネクション数を 12000 に設定しコンパイルしている。Solaris2.7 ではプロセスが開けるファイルディスクリプタの最大数が初期値で 1024 に設定されている。これを変更するため、/etc/system に次の 2 行を追加した。

```
set rlim_fd_max = 12800
set rlim_fd_cur = 12800
```

2.6 ircd の設定

2.6.1 ircd.conf の設定概要

ircd では、マシンや管理者の情報、クライアントに解放するポート番号など基本的な情報のほか、次の内容を設定する。

- 他のサーバとの接続
- それぞれのクライアントに許可する接続形態
- サービスを妨害する恐れがあるクライアントの拒否

ircd では接続先によって接続形態を分けることができる。これを接続クラスと呼ぶ。サーバとの接続クラスとクライアントの接続クラスは分ける。クラスを用いて、各サーバ、クライアントに対し最適な接続形態をとる。

2.6.2 サーバとの接続

irc.fujisawa は、現在 irc.tokyo.wide.ad.jp のリーフサーバとして IRCnet に接続している。サーバ間接続では、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する。また、接続クラスを指定し次のパラメータを設定する。括弧内は irc.fujisawa における値である。

- ping の頻度 (90 秒)
- 接続確認の頻度 (90 秒)
- send queue の大きさ (8000000bytes)

現在のバージョンでは、サーバ同士は IRC 網全体が木構造になるように接続される。そのため、サーバ間で新たに接続する場合どちらが上流でどちらが下流かをはっきりさせておくことが望ましい。今回の場合、irc.tokyo.wide.ad.jp が上流、irc.fujisawa が下流である。自動接続は下流から上流へと設定する。

2.6.3 クライアントの接続

irc.fujisawa は現在準備段階にあり、一般ユーザに対する接続サービスは行っていない。そのため現在は、WIDE 内部からの接続要求だけを許可している。しかし、IRCnet では国際的、国内的にサーバのユーザ受け入れに関するルールが存在し、本格的な運用を開始した後はこのルールに沿ってユーザを受け入れる方針である。

まず、国際的にはサーバを立ち上げている TLD のサーバ群は、自分の TLD 内のユーザを収容しなくてはならない。また、国内的ルールとして、各サーバの受け入れ可能なユーザ数のうち半数以上は国内のユーザを受け入れて収容できるようにする、というルールがある。国内的ルールは、国際的ルールと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生まれている。

WIDE 内の IRC サーバ群は次のファイルを共有し、国内向けに積極的なサービスを行なっている。

- 国内のドメイン名リスト
- 国内の ISP に割り当てられている IP アドレス空間のリスト
- 過去に、サービスを妨害する行動の踏台になったドメイン名、ホスト名のリスト

これらのリストは ircd の設定ファイルの一部として記述されており、設定ファイルに取り込んで使用する。ファイルの共有は CVS で行なっている。国内の ISP に割り当てられている IP アドレス空間のリストは、WIDE 内のルータの経路表から得ている。

クライアントにおけるクラスのパラメータには、次の項目がある。

- ping の頻度
- そのクラスのクライアントの最大受け入れ数
- send queue の大きさ
- 1 つのホストからローカルサーバへ接続できるクライアント数
- 1 つのホストから IRC 網へ接続できるクライアント数

WIDE では、国内のユーザが最大 2 クライアントまでサーバ (もしくは IRC 網) に接続できる設定を行なっている。しかし、NAT 越しのユーザは IP アドレスが同じになるため、NAT を用いているドメインに対してはこの制限を緩めている。また、国内の逆引きが設定されていない IP アドレスを持つクライアントに対しては、nickname が変更できない、チャンネルオペレータになれないという制限を付けて接続を許可する。

サービスを妨害する恐れがあるクライアントに対しては、接続を許可するべきではない。IRC ではサーバへのクラックのほか、チャンネル荒しなどサービスそのものを妨害する攻撃が存在する。これらの行為の多くは、既にクラックを受けたマシンや管理が行き届いていないマシンを踏台にして行なわれる。そのため、過去に悪意を持った行動の踏台にされたマシンはリストされ、ここからの接続を拒否している。

2.7 計測

藤沢 NOC に irc サーバを設置するにあたり、ネットワークに与える影響を調べるため、サーバマシンに関連するトラフィック量を計測した。また、現在のマシンパワーで ircd がどの程度のリソースを消費するのかを調べるため、CPU の使用率とメモリの使用量を計測した。

2.7.1 計測環境

データの測定は、irc.tokyo にリーフサーバとして接続し IRCnet に繋がった状態で行った。サーバに接続しているクライアント数は管理用の 2 つのみである。各クライアントは数個のグローバルチャンネルに JOIN している。

2.7.2 通信トラフィック

トラフィックデータとして、サーバマシンが接続しているスイッチにおける、ポートごとのトラフィックカウンタを SNMP を利用して収集した。はじめは 1 分間隔でデータを収集していたが、上流サーバに接続した直後に流れてくるトラフィックをより細かく測定するために 5 秒間隔の収集に変更した。これで得られた値を rrdtool を用いてデータベース化し、グラフとして出力した。

図 2.1 に、irc.tokyo とサーバ間接続する前後 5 分の測定結果を示す。接続を開始した直後に irc.tokyo から 3MBytes 強のバーストトラフィックが流れ、その後安定状態になっている。また、3MBytes という数値は irc.tokyo の &NOTICES チャンネルから得られたものである。

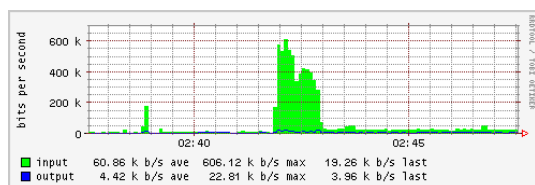


図 2.1 IRCnet に接続する前後 5 分間におけるトラフィックの変化

irc.fujisawa は単一のサーバとしか接続しないリーフサーバである。そのため、IRC サーバに関連するトラフィック量は、irc.fujisawa に接続しているクライアントの数と、各クライアントが JOIN している

チャンネルのメッセージ量が大きく影響する。今回の計測環境では、これらの要素が微少であった。そのため、図 2.2 に示すように、時より 300kbps 程度のトラフィックがあるが、出入り合計で平均して 70kbps 程度の定常トラフィックであった。

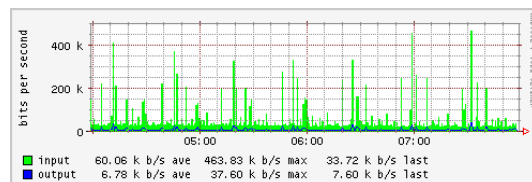


図 2.2 定常トラフィック

2.7.3 CPU とメモリの使用率

ircd 起動後 24 時間の、CPU の使用率の変化を図 2.3 に示す。同じくメモリの使用量の変化を図 2.4 に示す。

CPU の使用率は 1% から 5% 程度であり、それほど高くはない。しかし、クライアント数が少ない状態でも、ある程度の CPU リソースを消費している。

メモリの使用量は起動してから徐々に上昇し 60MBytes 程度まで増加した。その後はこの使用量で安定したが、マシン全体のメモリ量が 96M であることから、クライアント数が増加した場合にメモリ不足が起こらないよう、今後慎重に対処する必要がある。

2.8 今後の予定

現在 irc.fujisawa は学生によって運用されており、今回の活動にあたって、菊地 高広氏をはじめ IRCnet-jp の多くの方々に御助言を頂いた。今後は管理者の技術レベルの向上を計ると共に、藤沢 NOC における IRC サーバの運用経験を文書として蓄積していく予定である。

また、現在 irc.fujisawa は村井研究室内部のユーザにサービスを行なっているが、今後は徐々に多くのユーザにサービスを提供していく予定である。しかし、現在の計算機資源でどの程度のクライアント数までサービスを提供できるかは不透明であり、システムが破綻しないよう十分に監視していく必要がある。また、大規模なサービスに耐える計算機資源をできるだけ早く確保する必要がある。

その他には、藤沢 NOC の 6bone を利用した IPv6 によるサービスなどを予定している。

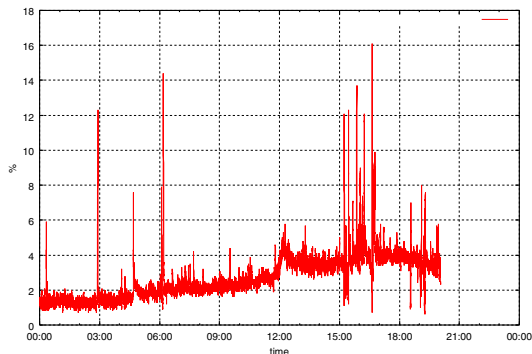


図 2.3 CPU の使用率の変化

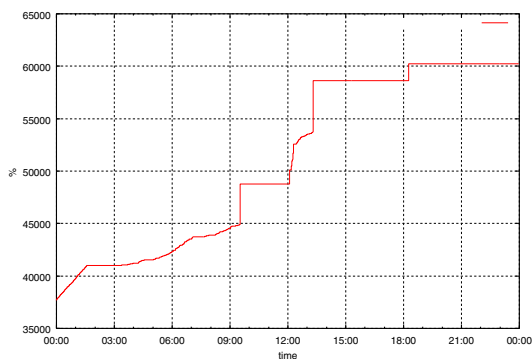


図 2.4 メモリの使用量の変化

第 3 章 IRC サーバの実装状況と日本語対応

現在 IRC サーバの実装は多くのものが存在するが、ここでは、12 年前の IRC 開発当初から続いている本家であり、かつ、IRCnet を始めとして世界中で幅広く用いられている実装について言及する。

3.1 irc2.10.3

現時点での最新版は 1999 年 8 月にリリースされた irc2.10.3 であり、1998 年にリリースされた irc2.10.0 からバグ修正や改善などが行なわれたものである。この版は Christophe Kalt 氏が中心となって開発が行なわれ、ircd-dev@irc.org メーリングリストがそれを支えている。我々も運用テストやバグ修正を送る形で参加しており、IPv6 対応やデバッグにお

いても協力してきている。

なお、2000 年 5 月にバグ修正版として irc2.10.3p1 がリリース予定である。irc2.9.x および irc2.10.x と実装をまとめてきた Christophe Kalt 氏は 2000 年 2 月に引退をし、Piotr Kucharski 氏に引き継がれている。

3.2 iauth

iauth は IRC サーバへの接続をしようとするクライアントの認証などを行なうため、IRC サーバ本体の ircd からスレーブプロセスとして起動される。ircd とプロセスが分離されている理由としては、次の点が挙げられる。

- ircd 本体と非同期に動作することができる。
- ircd 本体をこれ以上複雑にせず、きれいに機能を分離することができる。
- iauth のプログラムは複数のモジュールを任意に組み合わせて成り立っている。

最後の点のモジュールとは、認証などのために各プロトコル別にそれぞれ対応するモジュールが存在しており、新たに組み入れたい場合に容易に拡張することができるようにしくみとなっている。また、コンパイル時点で不要なものを組み入れないこともできる。

rfc931 モジュールは、rfc931[208]、後に更新されて rfc1413[209] で定められている、いわゆる ident を行なうモジュールである。この機能は irc2.9.x までは ircd 本体に組み込まれていたが、irc2.10.x から iauth に分離されている。

socks モジュールは、接続しに来ているクライアントのホストにおいて、いわゆる open socks server が動いていないかを調べるモジュールである。これは、4 節で述べる open socks 問題に対応するためのものであり、該当するホストからのクライアントの接続を拒否することができる。

我々は iauth についてもデバッグングにおいて開発の協力を行なってきた。それらの中に、socks モジュールにおける特に深刻なバグ修正があり、これは irc2.10.3 ではまだ反映されていないため、socks モジュール使用の際にはパッチを当てて運用する必要があると注意を要する。これについては 2000 年 5

月にリリース予定の irc2.10.3p1 で修正される予定である。

3.3 IRC に関する RFC

IRC の仕組みやプロトコルなどを定めた RFC としては、これまで 1993 年 5 月に発行された RFC1459[210] があった。これは、ステータスが experimental であり、1993 年 3 月にリリースされた irc2.8 を元にして記述されている。しかし、その後の irc2.9 や irc2.10 で機能やプロトコルが拡張されており、現在幅広く用いられている実装とは大きく乖離してしまっていた。

表 3.1 IRC に関する RFC 一覧

RFC1459 Internet Relay Chat Protocol
RFC2810 Internet Relay Chat: Architecture
RFC2811 Internet Relay Chat: Channel Management
RFC2812 Internet Relay Chat: Client Protocol
RFC2813 Internet Relay Chat: Server Protocol

今回、RFC1459 を更新する形で RFC2810[211], RFC2811[212], RFC2812[213], RFC2813[214] と、四つの RFC が 2000 年 4 月に発行された。ステータスはどれも informational であり、これらは最新の irc2.10.3 の状況を反映している。

3.4 IRC での日本語使用の問題

ここでは、IRC における日本語の符号化の問題、および、IRC において ISO-2022-JP を用いる場合の問題を整理し、これらの問題のいくつかを解決する IRC サーバの実装である jp 版について述べる。

3.4.1 日本語を用いる上での問題点

日本語において漢字をコンピュータ上で用いる場合の符号化の方法は様々であるが、IRC において漢字を用いる場合は特に以下の点に留意する必要がある。

- IRCnet など世界規模の IRC 網における各文字の符号化をどうするか。
- メールなどと違って、IRC のような短いメッセージやさらに短いチャンネル名においては符号化の方法を指定するのが困難である。
- チャンネル名を指定する時に符号化したものが

お互いに完全に一致しなければ同じチャンネルを利用することができないため、見ためは同じものが複数の表現をとらないようにしなければいけない。

日本において IRC が普及し始めた 1990 年 1 月に、当時はいわゆる JIS か EUC か SJIS かという状況において、国際的な環境を考慮していわゆる JIS を用いることが合意された。その後、チャンネル名での一致問題から、“ESC \$ @” と “ESC (J” は用いずに、“ESC \$ B” と “ESC (B” のみを用いることとなった。また、これらの切替えシーケンスが冗長に使われた場合も一致問題が発生するため、冗長な使用は制限される。ちなみに ISO-2022-JP は 1993 年 7 月に RFC1468[215] となっている。

3.4.2 ISO-2022-JP を用いる上での問題点

IRC において ISO-2022-JP を用いる場合には、次のような点から使用の際に問題が発生する。

- プロトコル上でチャンネル名の区切りに “,” を用いている。
- チャンネルの伝播範囲を指定するチャンネルマスク部分との区切りに “:” を用いている。
- 文字長が短く制限されているものは、自動的に各最大文字長へと短く切り捨てられてユーザへ伝えられる。

一つ目の問題は、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たくさんあるが、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して解釈され、意図したとおりのチャンネル名を作ることができない。これは、作成することができないチャンネル名を代表して “#がが” 問題と呼ばれることがある。

二つ目の問題は同様に、ISO-2022-JP においては “ず” や “歳” などをはじめとして “:” を潜在的に含む文字がたくさんあるが、これらの文字をチャンネル名に用いると “:” 以降がチャンネルマスクと解釈され、ほとんどの場合は自分の用いているサーバのサーバ名とチャンネルマスクが合致しないために意図したとおりのチャンネル名を作ることができない。これは “#ずず” 問題と呼ばれることがある。ちなみに、チャンネルマスク部分の判定は後ろから行なわれるため、“#ず

ず:*.jp”と有効なチャンネルマスクを付加することで“ず”を含むチャンネル名を使うことができる。

三つ目の問題は、文字長制限により自動的に切り捨てられることにより ISO-2022-JP のエスケープシーケンスが失われる場合に発生する。

3.4.3 クライアント側で対応が必要な場合

これら三つの問題のうち最後のものは、そのメッセージを受け取ったクライアントがそれを考慮して適切に処理をすれば解決する種類のものであるが、現存する日本語対応のクライアントの実装においては多くが完全には処置できていない。

一番やっかいなケースとして、IRC 上の whois の返答の例がある。whois の返答は、順にスペースで区切られて、ニックネーム、ユーザ名、ホスト名、名前等、と続く。このうちニックネームとホスト名は現在アルファベットや数字などであることが保証されているが、ユーザ名部分は制限が設けられていない。また、ユーザ名部分は 9 バイトまでと定められているため、ここへそれより長い漢字の文字列を指定している場合は、最後の ASCII コードへの切替シーケンスが欠落してクライアントへ返答されうる。

ほとんどの日本語対応クライアントの実装では、サーバから送られてきたメッセージをまず内部文字コードに変換をし、その後でメッセージ解析を行っている。すると、whois の返答においてユーザ名部分で ASCII への切替シーケンスが欠落している場合、そのままその後のホスト名部分も漢字が続いていると誤認してしまうことになる。よって、ユーザへ間違った表示を行ないうる。

whois の返答の場合は、ユーザ名とホスト名の間がスペースで区切られているため、内部文字コードに変換する際にスペースのあとは ASCII へ戻すなどの対処で対応できる。しかし、チャンネルへの join や会話の各メッセージには、先頭に必ず nickname!username@hostname がユーザ情報として付加されているためその部分の解析に失敗してしまい、該当するユーザの join や会話などを処理したり表示することができないなどの症状が出てしまう。

emacs 上で動く IRC クライアントである irchat においても昔は同様の症状が出ていたが、この問題に対応するために実装を全面的に見直して対処した。MS-Windows など動く IRC クライアントのほとんどの実装は、最初にサーバから受け取ったメッセー

ジを内部文字コードのいわゆる SJIS へと変換してしまうようで、この問題に対応できていない。

3.4.4 サーバ側で対応が必要な場合

一方、3.4.2 節の三つの問題のうちの一つ目と二つ目はクライアント側では対応することができない。また、プロトコル上の区切り文字という問題であるため、サーバ側で対応をするにしても一筋縄ではうまくいかない。そこで、調査検討した結果、次のような基本方針でいくことにした。

- “,” や “:” が漢字部分に現れても、これを区切り文字としては扱わない。
- この対応をしたサーバは、サーバ間接続の際に最初にそのことを相手へ伝える。
- “,” や “:” が漢字部分に現れるチャンネルの情報は、相手のサーバが同じ対応をしたサーバである時のみ、その情報を相手へ伝播させる。

このような方法で実装することにより、該当するチャンネルはチャンネル情報の伝播という観点からちょうどマスク付チャンネルと同じ位置付けになるため、同じ IRC 網内に非対応サーバとの接続があっても相互接続上のプロトコルや非対応サーバに影響がないように実現することが可能となった。

また、三つ目の問題の中心であるユーザ名問題についても、きちんと対応できているクライアントの実装が少ないという状況を考慮してサーバ側で漢字を含むようなユーザ名を受け付けないという形で対応している。

これらの日本語問題のための対策を行なったサーバの実装を jp 版と呼び、現在の最新版は irc2.10.3 に対応した irc2.10.3+jp6 である。この jp 版は公開しており国内の多くのところで用いられている。これにより、IRC ではいままでチャンネル名に使用することができなかった、あるいは、使用において制限されていた数百文字の漢字をすべて制限なく使用することができるようになっている。

第 4 章 IRC の利用状況と分析

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

その状況と問題を述べる。

4.1 全体状況

インターネット上には数多くの IRC 網があるが、そのうち世界最大規模であるとともに最も古く、IRC 黎明期当初から続いているのが IRCnet である。現在約 130 のサーバが接続しており、ユーザの多い時間帯ではクライアント数 65000 以上、チャンネル数 30000 以上が存在している。

現在 IRCnet のサーバが上がっているところは 29 カ国あり、at, au, be, ca, ch, cz, de, dk, ee, fi, fr, gr, hr, hu, il, is, it, jp, lv, nl, no, pl, ru, se, si, sk, tw, uk, us といった国々で構成されている。それぞれの国によって抱えるサーバ数やユーザ数の規模は様々であり差が大きい。

4.2 国際接続

基本的に各国それぞれにおいて国内のサーバ同士が繋がってまとまっており、それが国際的なレベルのハブサーバへ接続する、といった感じになっている。このとき、ほとんどの国では国外のサーバに対してサーバ名を集約して見せかけることで一つのサーバだけが存在しているように見せている。例えば、トップレベルドメインが xx である国がサーバ名が xx で終わる複数のサーバを持っていた場合、対外的には全体として*.xx という名前の一つのサーバであるように見せる。

日本の場合も全体として*.jp というサーバ名で国外に対して名乗っている。接続先は US にある国際的ハブサーバである ircd.stealth.net と ircd.webbnet.net であり、国内のハブサーバである irc.tokyo.wide.ad.jp と race-server.race.u-tokyo.ac.jp からそれぞれ WIDE の海外線と SINET の海外線を用いて接続していたが、どちらも満足なパフォーマンスが出ない状況が時々あって安定性を欠いていた。現在は、apan.net および nordu.net 経由でフィンランドにある国際ハブサーバ ircd.funet.fi との接続も持つことによって、安定するとともに会話の遅延現象なども解消されている。一時期、マレーシアの irc.inter.net.my とも接続を持っていたが、マレーシアのサーバは DoS により回線を何度も潰されたため廃止された。

4.3 国内状況

2000 年 3 月の時点での国内の IRCnet のサーバは表 4.1 のようになっている。

表 4.1 IRCnet の国内のサーバ

サーバ名	開放ポート	備考
akiu.gw.tohoku.ac.jp	6667	
irc.dti.ne.jp	6666 - 6667	
irc.huie.hokudai.ac.jp	6667	
irc.karrn.ad.jp	6660 - 6669	2000 年 3 月に廃止
irc.kyoto.wide.ad.jp	6660 - 6669	
irc.rcac.tdi.co.jp	6660 - 6669	
irc.tokyo.wide.ad.jp	6660 - 6669	
irc6.rcac.tdi.co.jp	6667	IPv6 のみ
race-server.race.u-tokyo.ac.jp	6667	2000 年 3 月に廃止

このうち、irc6.rcac.tdi.co.jp のみ IPv6 でサービスを行っており、その他は IPv4 で行っている。また、どのサーバも 3.4.4 節で述べた irc2.10.3+jp6 を用いている。

現在はこれらに加えて第 2 章で述べた irc.fujisawa.wide.ad.jp が接続されており、一般ユーザへの公開も予定されている。また、京都大学、東京大学、早稲田大学それぞれにおいても近いうちに IRC サーバが復活して IRCnet へ再び接続される計画となっている。

4.4 問題ユーザとサービスの妨害

IRCnet は規模が大きいとともに各チャンネルで多くの世界が作られているため、その IRC の世界の中でチャンネルの乗っ取りをはじめとする様々なよくない行為もたくさん行なわれている。これらのうちには他人のホストを踏台にして IRC サーバに接続してよくない行為を行なう者が多く、大きな問題となっている。

そのような踏台行為のうち、初心者でも簡単に行なうことができるものとして世界中に非常に多く存在している open socks server を利用するものがある。この open socks server とは、socks[216] の設定がきちんとなされていないために外から誰でも自由に使用することができてしまう状態のホストを指す。これを利用した悪用があまりにも多いとともに、open socks server をブラックホストへと登録していても次々と新たに現れてしまう状況のため、3 節で述べたように open socks server から IRC サーバへのクライアントの接続を拒否する機能が装備された。

また、セキュリティが甘くて簡単に侵入できてしまうホストが世界中に多く存在しているため、それ

らに侵入して eggdrop などの常駐型の IRC クライアントを寄生させて踏台として自由に用いることも多くなっている。さらに、侵入先で root 権限をとることができた場合は、そのマシンが接続している LAN の中で用いられていない IP アドレスをすべて勝手にそのマシンから利用できるようにしてしまうとともに、それらの多くの IP アドレスを使って大量に IRC サーバへ接続を行なって悪さをする mirkforce というプログラムも最近では使われることが多くなってきている。

一方、チャンネルやニックネームの乗っ取りなどの目的のため、IRC サーバ間の切断などを狙って IRC サーバへの DoS など非常に頻繁に行なわれており、DDoS がひろまった現状ではその激しさも増し、我々のサーバも smurf やその他で数十 Mbps の攻撃を受けたりしている。

このため、IRC だけでなく帯域全体を真っ黒にすることで被害が深刻なため、4 節で述べたように IRC サーバが廃止に追い込まれた国や組織も多い。このため、これらの妨害行為に抗議して IRCnet のうちの 19 カ国が参加して 2000 年 4 月 7 日から 8 日にかけてストライキが行なわれるなどの状況になっているが、根本的な解決には至っていない。

第 5 章 IRCnet の現状と問題

5.1 各時刻のクライアント数の分析

WIDE IRC WG では 1999 年度に irc.tokyo.wide.ad.jp, irc.kyoto.wide.ad.jp, irc.race.u-tokyo.ac.jp, irc.rcac.tdi.co.jp の 4 台の IRC サーバを運用し、毎時の接続数と接続元 IP アドレスを記録した。国内の IRC サーバには他に irc.karrn.ad.jp, irc.tohoku.ac.jp, irc.huie.hokudai.ac.jp, irc.dti.ne.jp があるが、データを取得できた WIDE 内のサーバの利用者数のみ論じることとする。データは 1999 年 4 月 12 日から 2000 年 4 月 18 日までのものを使用した。

一日の接続数の最大値をプロットしたのが図 5.1 である。これが年間のクライアント数の変化となる。

図 5.1 によると、ユーザ数は一年間で 5000 人から 12000 人までほぼ直線的にのびていることがわかる。

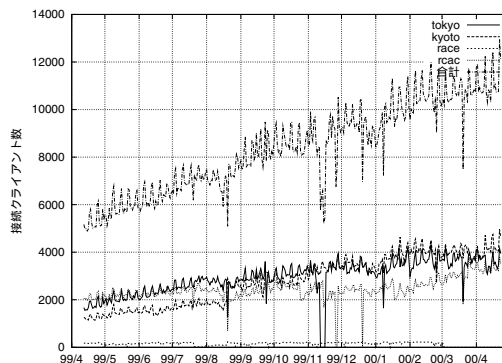


図 5.1 1999 年度のクライアント数の変化

irc.race.u-tokyo.ac.jp は収容可能人数が少なかったためすくない。他の三台では、irc.kyoto.wide.ad.jp と irc.tokyo.wide.ad.jp に比べ、irc.rcac.tdi.co.jp が少なめであるが、最近ではよく使われるようになってきている。また、システムの故障やメンテナンス、DoS アタックなどでサーバが停止するとそれがそのまま全体の利用者数の減につながっている。接続できないとその日の接続をあきらめてしまっている可能性がある。

次に、年間を通して週の各時刻ごとの平均をとり、プロットしたものが図 5.2 である。

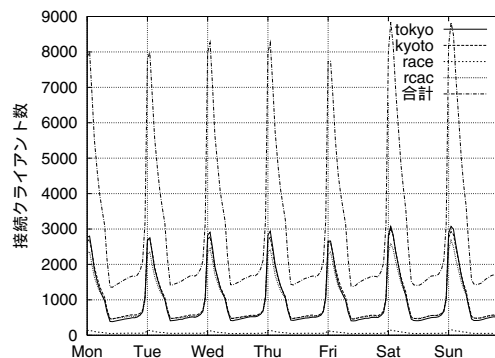


図 5.2 1999 年度の週間のクライアント数の変化

図 5.2 によると、曜日ごとの変化は少ないが、土日の夜に 1000 人ほどユーザ数が多くなるようである。図 5.1 での周期的な増減はこれである。時間ごとの変化をみると、昼間がいちばん少ないがそれでも 1500 人程度ユーザがいて、22 時ごろからユーザが急増し、テレホーダイ時間帯の 1 時に最大となり、テレホーダイ時間終了とともに減るような利用状況となっている。

次に、IRC サーバへの接続元 IP アドレスから AS 番号を得て各 AS からの接続数を調べ、一年間の平均をとって、一日の各時刻ごとの上位 10 AS を調べたのが表 5.1 である。図 5.2 にて利用が最小になる昼間には、OCN や ODN のような個人向け専用線接続ユーザが多い AS や RIM のような個人にシェルログインサービスを提供している AS、SINET、WIDE などの大学系が目立つ。ところが夜になるとダイヤルアップユーザが増えるようで、ODN が突出し、OCN もユーザ数が倍になり、朝 7 時まではダイヤルアップユーザを多く抱えている AS で上位 10 位を占める。

IRC サーバごとのユーザの接続元 AS を調べた表が表 5.2 である。これは、一年間の全日の 24 時間分のデータをサーバごとに積算し、AS ごとのユーザ数の平均値を求めたものであるため、常時接続者の傾向が強めにでる。この表により、どの AS からの利用者がどのサーバを好んで使うかをみることが出来る。表によると、きわだった差はないようであるが、irc.kyoto.wide.ad.jp と irc.rcac.tdi.co.jp に OMP や NCA5 のユーザが多いようである。ユーザが近いことを選んで選んでいるようである。

参考のために 1999 年 4 月と 2000 年 4 月の一週間のクライアント接続数のグラフを図 5.3 と図 5.4 に示す。変化パターンは同じで接続クライアント数だけ倍になっている。

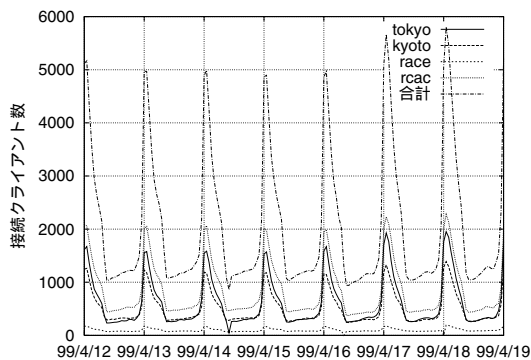


図 5.3 1999 年 4 月の一週間のクライアント数の変化

これからも利用者が増える傾向にあるため、現在の一般的な公開サービスとして IRC を普及させていくなれば、収容可能人数を増やしていく必要がある。また、常時接続利用者も同じような割合で増えていて、これからさらに増える可能性がある。

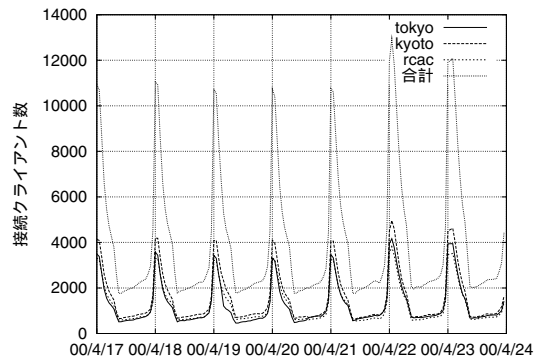


図 5.4 2000 年 4 月の一週間のクライアント数の変化

5.2 クライアントの接続頻度の分析

次に、クライアントの接続開始時刻データを分析し、ユーザが接続し始める時刻を調べた。そして、ユーザの接続開始時間をもとに、毎分に接続を開始したクライアント数を求めた。これについては、irc.tokyo.wide.ad.jp , irc.kyoto.wide.ad.jp , irc.rcac.tdi.co.jp の三台の IRC サーバのユーザ接続データをもとに、三台のデータを融合して分析した。データは 1999 年 12 月 15 日から 2000 年 5 月 6 日までのものを使用した。

まず、期間中の一日のデータのうちの最大値をプロットしたものが図 5.5 である。特に大きくなっている日はネットワーク障害やサーバの再起動などがあり、接続が多くなっているため、それ以外をみると、正月と 5 月のゴールデンウィークですこし頻度がさがっているが、毎日一分間に 400 クライアント程度の接続が来る時間があることがわかる。また、すこし増加傾向にあるようにみえる。

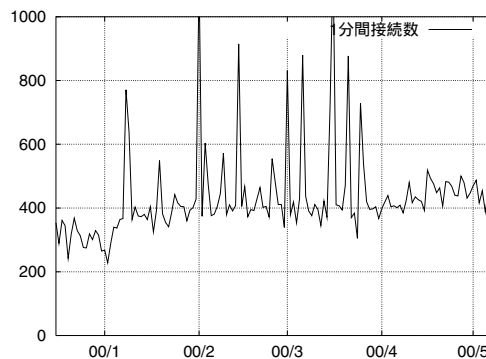


図 5.5 クライアント接続開始数 (日の最大値) の変化

第 17 部 IRC の運用技術と活用技術

表 5.1 1999 年度の時間ごとのクライアント接続元の変化

時刻	0 時		1 時		2 時		3 時	
	AS	clients %	AS	clients %	AS	clients %	AS	clients %
1	ODN	1118 14.30	ODN	1169 14.35	ODN	1026 14.24	ODN	824 13.84
2	OCN	682 8.73	OCN	692 8.50	OCN	632 8.77	OCN	557 9.35
3	IJ	429 5.49	IJ	443 5.44	IJ	389 5.40	IJ	314 5.28
4	DTI	404 5.17	DTI	424 5.21	DTI	363 5.05	DTI	287 4.83
5	SPIN	365 4.67	SPIN	392 4.81	SPIN	347 4.82	SPIN	278 4.67
6	SINFONY	352 4.51	SINFONY	376 4.62	SINFONY	320 4.44	SINFONY	247 4.16
7	INFOWEB	292 3.74	INFOWEB	299 3.67	KCOM	246 3.42	KCOM	208 3.50
8	KCOM	254 3.25	KCOM	270 3.32	INFOWEB	241 3.35	DOLPHIN	181 3.04
9	DOLPHIN	239 3.07	DOLPHIN	254 3.13	DOLPHIN	223 3.11	INFOWEB	178 2.99
10	OMP	223 2.86	OMP	228 2.80	OMP	201 2.79	OMP	167 2.81
合計		7819		8151		7210		5956
時刻	4 時		5 時		6 時		7 時	
	AS	clients %	AS	clients %	AS	clients %	AS	clients %
1	ODN	660 13.37	ODN	543 12.87	ODN	459 12.38	OCN	406 12.50
2	OCN	498 10.09	OCN	458 10.86	OCN	430 11.61	ODN	380 11.70
3	IJ	255 5.18	IJ	215 5.10	IJ	185 5.01	IJ	161 4.95
4	DTI	227 4.60	DTI	183 4.36	DTI	152 4.10	DTI	124 3.83
5	SPIN	219 4.43	SPIN	177 4.20	SPIN	147 3.99	SPIN	123 3.79
6	SINFONY	190 3.86	KCOM	153 3.63	KCOM	136 3.67	SINET	121 3.74
7	KCOM	177 3.59	SINFONY	150 3.57	SINET	128 3.47	KCOM	118 3.63
8	DOLPHIN	146 2.97	SINET	135 3.21	SINFONY	123 3.33	SINFONY	99 3.06
9	SINET	145 2.94	DOLPHIN	123 2.92	DOLPHIN	107 2.89	DOLPHIN	92 2.85
10	OMP	138 2.81	OMP	119 2.83	OMP	105 2.84	OMP	92 2.85
合計		4940		4220		3709		3253
時刻	8 時		9 時		10 時		11 時	
	AS	clients %	AS	clients %	AS	clients %	AS	clients %
1	OCN	354 15.59	OCN	311 21.75	OCN	315 22.67	OCN	321 22.69
2	ODN	218 9.63	SINET	105 7.35	SINET	106 7.67	SINET	110 7.83
3	SINET	113 4.99	ODN	77 5.40	WIDE	69 5.00	WIDE	70 4.99
4	IJ	109 4.81	IJ	71 5.00	IJ	69 4.97	IJ	70 4.98
5	WIDE	69 3.04	WIDE	68 4.77	RIM	68 4.92	RIM	69 4.91
6	RIM	68 3.01	RIM	67 4.71	ODN	65 4.69	ODN	64 4.56
7	SPIN	67 2.99	DDI	45 3.15	DDI	46 3.37	DDI	49 3.47
8	DTI	67 2.98	TTNET	39 2.75	TTNET	38 2.76	TTNET	38 2.72
9	SINFONY	64 2.82	OMP	38 2.72	OMP	37 2.69	OMP	37 2.67
10	KCOM	62 2.74	NCA5	34 2.44	NCA5	35 2.55	NCA5	36 2.58
合計		2273		1430		1392		1415
時刻	12 時		13 時		14 時		15 時	
	AS	clients %	AS	clients %	AS	clients %	AS	clients %
1	OCN	322 22.42	OCN	325 22.12	OCN	328 21.68	OCN	332 21.31
2	SINET	114 7.96	SINET	119 8.08	SINET	122 8.09	SINET	126 8.14
3	IJ	73 5.10	IJ	75 5.14	IJ	79 5.28	IJ	83 5.35
4	WIDE	71 4.95	WIDE	72 4.92	WIDE	73 4.87	ODN	76 4.93
5	RIM	69 4.82	RIM	69 4.72	ODN	71 4.74	WIDE	74 4.78
6	ODN	65 4.54	ODN	67 4.58	RIM	69 4.59	RIM	69 4.46
7	DDI	49 3.46	DDI	50 3.43	DDI	50 3.36	DDI	51 3.29
8	TTNET	39 2.73	TTNET	40 2.74	TTNET	41 2.74	TTNET	42 2.73
9	OMP	38 2.67	OMP	38 2.64	OMP	39 2.62	OMP	40 2.61
10	NCA5	36 2.57	NCA5	38 2.59	NCA5	38 2.58	NCA5	39 2.55
合計		1436		1473		1513		1558
時刻	16 時		17 時		18 時		19 時	
	AS	clients %	AS	clients %	AS	clients %	AS	clients %
1	OCN	335 20.99	OCN	340 20.75	OCN	338 20.47	OCN	334 20.33
2	SINET	129 8.14	SINET	132 8.06	SINET	131 7.96	SINET	129 7.87
3	IJ	86 5.39	IJ	89 5.34	IJ	92 5.57	IJ	95 5.82
4	ODN	82 5.15	ODN	87 5.46	ODN	92 5.57	ODN	93 5.68
5	WIDE	74 4.68	WIDE	75 4.63	WIDE	76 4.60	WIDE	75 4.57
6	RIM	69 4.34	RIM	69 4.26	RIM	69 4.20	RIM	68 4.18
7	DDI	52 3.26	DDI	52 3.23	DDI	52 3.17	DDI	50 3.06
8	TTNET	42 2.69	TTNET	44 2.69	TTNET	44 2.70	TTNET	44 2.70
9	OMP	41 2.61	OMP	42 2.62	OMP	43 2.65	OMP	44 2.68
10	NCA5	40 2.54	NCA5	40 2.49	NCA5	40 2.47	NCA5	40 2.45
合計		1597		1639		1651		1644
時刻	20 時		21 時		22 時		23 時	
	AS	clients %	AS	clients %	AS	clients %	AS	clients %
1	OCN	335 19.98	OCN	345 19.02	OCN	362 17.56	OCN	422 13.46
2	SINET	128 7.65	SINET	128 7.05	ODN	163 7.91	ODN	365 11.65
3	IJ	104 6.24	IJ	121 6.69	IJ	148 7.21	IJ	213 6.81
4	ODN	99 5.94	ODN	121 6.69	SINET	129 6.29	SINET	140 4.48
5	WIDE	74 4.44	WIDE	74 4.07	WIDE	74 3.60	DTI	126 4.03
6	RIM	68 4.09	RIM	68 3.78	RIM	68 3.33	SINFONY	118 3.77
7	DDI	49 2.97	OMP	51 2.84	SINFONY	64 3.13	SPIN	114 3.63
8	OMP	46 2.76	DDI	51 2.81	D'TI	60 2.94	OMP	89 2.84
9	TTNET	46 2.76	TTNET	50 2.80	OMP	59 2.88	TTNET	82 2.63
10	NCA5	39 2.38	SINFONY	47 2.62	TTNET	57 2.78	INFOWEB	80 2.55
合計		1678		1817		2062		3140

次に、期間中のデータを全体で平均し、一週間の変化を示したものが図 5.6 である。23 時ごろの突出以外でなだらかに変化していない部分は前述のイレギュラーなデータのために突出したものであると考えられる。このデータと図 5.2 のクライアント数の変化をあわせて見ると一週間のうちで金曜の夜と土

曜の夜は他の曜日よりも若干集中度が低いクライアント数は多いことがわかる。23 時前後に幅広く接続開始するようである。

次に、一日のあいだの接続数の変化を図テレホーダイ時間帯の 23 時前後に注目した図を図 5.8 に示す。これによると、23 時に一分間に 400 クライアン

表 5.2 1999 年度の IRC サーバごとのクライアント接続元

	irc.tokyo.wide.ad.jp		irc.kyoto.wide.ad.jp		irc.race.u-tokyo.ac.jp		irc.rcac.tdi.co.jp	
	AS	数 %	AS	数 %	AS	数 %	AS	数 %
1	OCN	120.7 11.91	OCN	144.0 13.77	OCN	16.4 19.11	OCN	136.3 14.15
2	ODN	119.0 11.74	ODN	124.6 11.91	SINET	8.3 9.65	ODN	90.8 9.43
3	IJ	62.6 6.18	SINET	54.9 5.25	TISN	5.5 6.43	IJ	51.5 5.35
4	DTI	41.1 4.06	HJ	51.3 4.90	ODN	5.4 6.27	SINET	38.5 4.00
5	SINFONY	39.5 3.89	OMP	44.5 4.25	RIM	4.8 5.55	DTI	33.7 3.50
6	SPIN	37.8 3.73	DTI	39.7 3.79	IJ	3.9 4.58	DDI	31.9 3.31
7	DOLPHIN	34.3 3.38	SPIN	39.2 3.75	TTNET	2.3 2.70	SINFONY	31.7 3.29
8	SINET	33.4 3.30	SINFONY	33.2 3.18	SINFONY	2.2 2.60	SPIN	30.5 3.17
9	KCOM	32.6 3.21	INFOWEB	26.8 2.56	DOLPHIN	1.6 1.92	RIM	26.6 2.76
10	TTNET	31.5 3.11	NCA5	23.9 2.28	OMP	1.6 1.91	DOLPHIN	26.0 2.70
11	INFOWEB	30.0 2.95	WIDE	22.8 2.18	DDI	1.6 1.81	OMP	24.6 2.55
12	WIDE	27.9 2.75	DDI	22.1 2.11	WIDE	1.5 1.76	KCOM	24.0 2.49
13	INTERVIA	24.5 2.42	RIM	21.7 2.07	SPIN	1.4 1.59	WIDE	23.4 2.43
14	NTTPC	23.2 2.29	KCOM	21.4 2.04	DTI	1.2 1.40	TTNET	22.3 2.32
15	HIGHWAY	21.8 2.15	TTNET	21.4 2.04	NTTPC	1.1 1.34	INFOWEB	21.7 2.26
16	RIM	19.9 1.96	INTERVIA	19.0 1.82	INFOWEB	1.1 1.31	NTTPC	15.9 1.65
17	TITUS	16.7 1.64	DOLPHIN	18.3 1.75	TOPIC	1.1 1.29	HIGHWAY	15.9 1.65
18	JOIN	16.1 1.59	HIGHWAY	17.0 1.62	KCOM	1.0 1.22	INTERVIA	15.1 1.57
19	DDI	15.9 1.57	NTTPC	16.4 1.57	HIGHWAY	1.0 1.19	NCA5	14.2 1.48
20	OMP	15.6 1.54	MESH	11.9 1.13	2554	1.0 1.15	TITUS	14.1 1.46
21	INTERQ	12.1 1.19	CTC	11.5 1.10	TITUS	0.9 1.11	IMNET	13.8 1.44
22	NIS	12.0 1.18	JOIN	11.1 1.06	INTERVIA	0.9 1.09	TOKYONET	13.2 1.37
23	MESH	12.0 1.18	TITUS	10.2 0.97	TOKYONET	0.8 0.95	JOIN	12.8 1.33
24	IMNET	10.5 1.04	TOKYONET	9.3 0.89	KARRN	0.8 0.92	ITNET	11.2 1.17
25	TOKYONET	10.5 1.04	INTERQ	9.0 0.86	PROX	0.8 0.88	MESH	10.2 1.06
26	AIR	8.9 0.88	QTNET	8.8 0.84	NIS	0.7 0.84	PROX	10.1 1.05
27	TISN	8.4 0.83	NIS	8.4 0.80	NCA5	0.7 0.84	NIS	9.7 1.01
28	MTCI	8.2 0.81	ODINS	7.6 0.73	CTC	0.7 0.83	CTC	9.0 0.93
29	ITNET	7.6 0.75	MTCI	6.8 0.65	ODINS	0.6 0.71	AIR	8.5 0.88
30	2554	7.5 0.74	3WEB	6.7 0.64	INTERQ	0.6 0.70	2554	7.3 0.76
合計		1013		1046		85		962

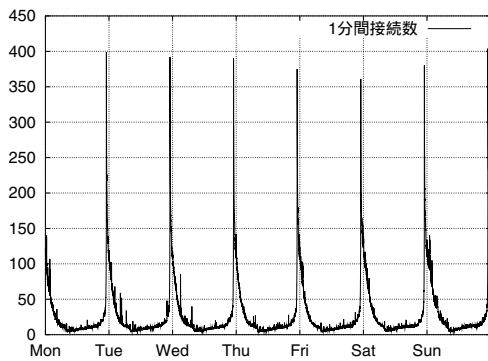


図 5.6 クライアント接続開始数の一週間の変化

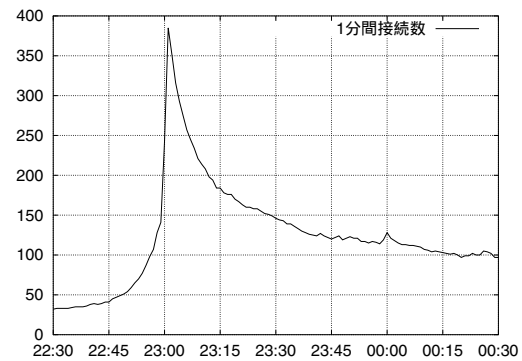


図 5.8 クライアント接続開始数の一日の変化 (テレホーダイサービス開始時刻付近)

ト程度の接続が集中し、そのあとなだらかに減っている。朝 8 時が最低になり、そのあと 22 時まで微増している。

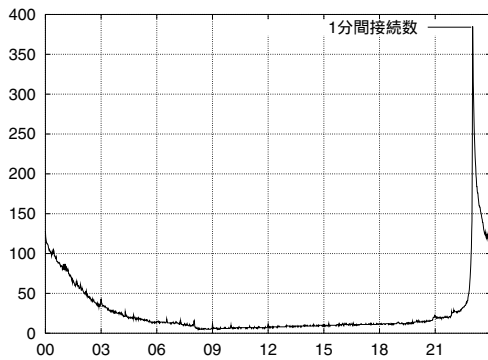


図 5.7 クライアント接続開始数の一日の変化

参考に 2000 年 4 月第三週の一週間の変化を図 5.9 に示す。図 5.6 と同じ変化をしている。

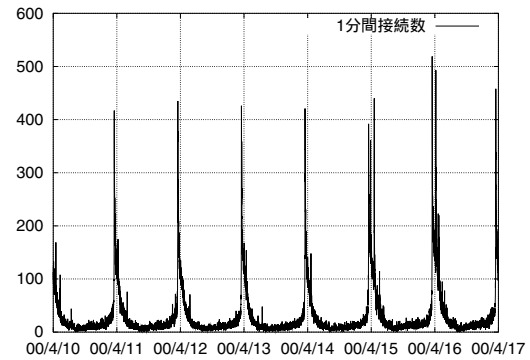


図 5.9 2000 年 4 月第三週のクライアント接続開始数の変化