

Network Working Group

Request for Comments: 2810

Updates: 1459

Category: Informational

C. Kalt

April 2000

Internet Relay Chat: Architecture

インターネットリレーチャット : アーキテクチャ

Status of this Memo

本文書の位置付け

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

このメモは、インターネットコミュニティに情報を提供します。いかなる種類のインターネット標準を指定しません。このメモの配布は無制限です。

Copyright Notice

著作権表示

Copyright (C) The Internet Society (2000). All Rights Reserved.

Copyright (c) The Internet Society (2000) 。全著作権所有。

Abstract

概要

The IRC (Internet Relay Chat) protocol is for use with text based conferencing. It has been developed since 1989 when it was originally implemented as a mean for users on a BBS to chat amongst themselves.

IRC（インターネットリレーチャット）プロトコルは、テキストベースの会議で使用するためのものです。1989年に、BBSのユーザーが自分自身の間でチャットするための平均として元々実装されたときから開発されてきました。

First formally documented in May 1993 by RFC 1459 [IRC], the protocol has kept evolving. This document is an update describing the architecture of the current IRC protocol and the role of its different components. Other documents describe in detail the protocol used between the various components defined here.

1993年5月にRFC 1459 [IRC]によって最初に正式に文書化されたこのプロトコルは進化し続けています。このドキュメントは、現在のIRCプロトコルのアーキテクチャとそのさまざまなコンポーネントの役割を説明する更新です。他のドキュメントは、ここで定義されているさまざまなコンポーネント間で使用されるプロトコルを詳細に説明しています。

Table of Contents

目次

1. Introduction	2
2. Components	2
2.1 Servers	2
2.2 Clients	3
2.2.1 User Clients	3
2.2.2 Service Clients	3
3. Architecture	3
4. IRC Protocol Services	4
4.1 Client Locator	4
4.2 Message Relaying	4
4.3 Channel Hosting And Management	4
5. IRC Concepts	4
5.1 One-To-One Communication	5
5.2 One-To-Many	5
5.2.1 To A Channel	5
5.2.2 To A Host/Server Mask	6
5.2.3 To A List	6
5.3 One-To-All	6
5.3.1 Client-to-Client	6
5.3.2 Client-to-Server	7
5.3.3 Server-to-Server	7
6. Current Problems	7
6.1 Scalability	7
6.2 Reliability	7
6.3 Network Congestion	7
6.4 Privacy	8
7. Security Considerations	8
8. Current Support And Availability	8
9. Acknowledgements	8
10. References	8
11. Author's Address	9
12. Full Copyright Statement	10

1. Introduction

The IRC (Internet Relay Chat) protocol has been designed over a number of years for use with text based conferencing. This document describes its current architecture.

The IRC Protocol is based on the client-server model, and is well suited to running on many machines in a distributed fashion. A typical setup involves a single process (the server) forming a central point for clients (or other servers) to connect to, performing the required message delivery/multiplexing and other functions.

This distributed model, which requires each server to have a copy of the global state information, is still the most flagrant problem of the protocol as it is a serious handicap, which limits the maximum size a network can reach. If the existing networks have been able to keep growing at an incredible pace, we must thank hardware manufacturers for giving us ever more powerful systems.

2. Components

The following paragraphs define the basic components of the IRC protocol.

2.1 Servers

The server forms the backbone of IRC as it is the only component of the protocol which is able to link all the other components together: it provides a point to which clients may connect to talk to

each other [IRC-CLIENT], and a point for other servers to connect to [IRC-SERVER]. The server is also responsible for providing the basic services defined by the IRC protocol.

2.2 Clients

A client is anything connecting to a server that is not another server. There are two types of clients which both serve a different purpose.

2.2.1 User Clients

User clients are generally programs providing a text based interface that is used to communicate interactively via IRC. This particular type of clients is often referred as "users".

2.2.2 Service Clients

Unlike users, service clients are not intended to be used manually nor for talking. They have a more limited access to the chat functions of the protocol, while optionally having access to more private data from the servers.

Services are typically automatons used to provide some kind of service (not necessarily related to IRC itself) to users. An example is a service collecting statistics about the origin of users connected on the IRC network.

3. Architecture

An IRC network is defined by a group of servers connected to each other. A single server forms the simplest IRC network.

1. はじめに

IRC（インターネットリレーチャット）プロトコルは、テキストベースの会議で使用するために長年にわたって設計されてきました。このドキュメントでは、現在のアーキテクチャについて説明しています。

IRCプロトコルは、クライアントサーバーモデルに基づいており、分散型ファッションで多くのマシンでの実行に適しています。典型的なセットアップには、クライアント（または他のサーバー）が接続するための中央ポイントを形成する単一のプロセス（サーバー）が含まれ、必要なメッセージ配信/多重化およびその他の機能を実行します。

各サーバーがグローバルな状態情報のコピーを持たせる必要があるこの分散モデルは、ネットワークが到達できる最大サイズを制限する深刻なハンディキャップであるため、プロトコルの最も重大な問題です。既存のネットワークが信じられないほどのペースで成長し続けることができた場合、ハードウェアメーカーがこれまで以上に強力なシステムを提供してくれたことに感謝しなければなりません。

2. コンポーネント

次の段落では、IRCプロトコルの基本コンポーネントを定義します。

2.1 サーバー

サーバーは、他のすべてのコンポーネントと一緒にリンクできるプロトコルの唯一のコンポーネントであるため、IRCのバックボーンを形成します。

お互い[IRC-Client]、および他のサーバーが[IRCサーバー]に接続するポイント。サーバーは、IRCプロトコルによって定義された基本サービスを提供する責任もあります。

2.2 クライアント

クライアントは、別のサーバーではないサーバーに接続するものです。どちらも異なる目的を果たす2つのタイプのクライアントがあります。

2.2.1 ユーザークライアント

ユーザークライアントは通常、IRCを介してインタラクティブに通信するために使用されるテキストベースのインターフェイスを提供するプログラムです。この特定のタイプのクライアントは、しばしば「ユーザー」と呼ばれます。

2.2.2 サービスクライアント

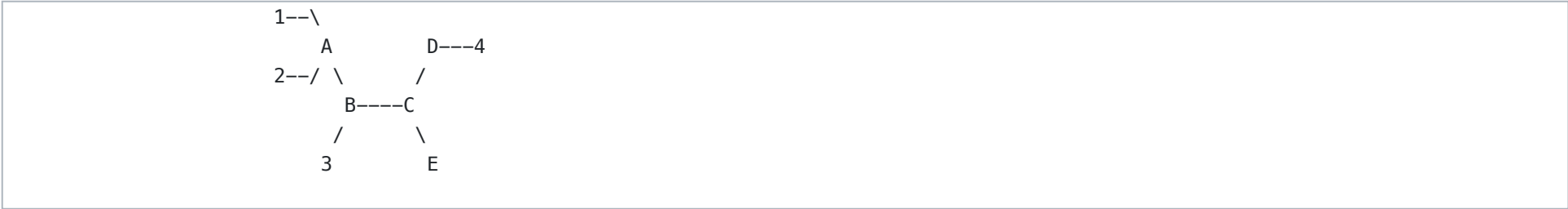
ユーザーとは異なり、サービスクライアントは手動で使用することも話したりすることを意図していません。プロトコルのチャット関数へのアクセスは、オプションでサーバーからより多くのプライベートデータにアクセスできます。

サービスは通常、ユーザーに何らかのサービス（必ずしもIRC自体に関連するわけではない）を提供するために使用されるオートマトンです。例は、IRCネットワークで接続されているユーザーの起源に関する統計を収集するサービスです。

3. 建築

IRCネットワークは、相互に接続されたサーバーのグループによって定義されます。単一のサーバーが最も単純なIRCネットワークを形成します。

The only network configuration allowed for IRC servers is that of a spanning tree where each server acts as a central node for the rest of the network it sees.



Servers: A, B, C, D, E Clients: 1, 2, 3, 4

[Fig. 1. Sample small IRC network]

The IRC protocol provides no mean for two clients to directly communicate. All communication between clients is relayed by the server(s).

4. IRC Protocol Services

This section describes the services offered by the IRC protocol. The combination of these services allow real-time conferencing.

4.1 Client Locator

To be able to exchange messages, two clients must be able to locate each other.

Upon connecting to a server, a client registers using a label which is then used by other servers and clients to know where the client is located. Servers are responsible for keeping track of all the labels being used.

4.2 Message Relaying

The IRC protocol provides no mean for two clients to directly communicate. All communication between clients is relayed by the server(s).

4.3 Channel Hosting And Management

A channel is a named group of one or more users which will all receive messages addressed to that channel. A channel is characterized by its name and current members, it also has a set of properties which can be manipulated by (some of) its members.

Channels provide a mean for a message to be sent to several clients. Servers host channels, providing the necessary message multiplexing. Servers are also responsible for managing channels by keeping track of the channel members. The exact role of servers is defined in "Internet Relay Chat: Channel Management" [IRC-CHAN].

5. IRC Concepts

This section is devoted to describing the actual concepts behind the organization of the IRC protocol and how different classes of messages are delivered.

5.1 One-To-One Communication

Communication on a one-to-one basis is usually performed by clients, since most server-server traffic is not a result of servers talking only to each other. To provide a means for clients to talk to each other, it is REQUIRED that all servers be able to send a

IRCサーバーに許可されている唯一のネットワーク構成は、各サーバーが表示されるネットワークの残りの部分の中央ノードとして機能するスパニングツリーの唯一のネットワーク構成です。

サーバー：A、B、C、D、Eクライアント：1、2、3、4

[図1.小さなIRCネットワークのサンプル]

IRCプロトコルは、2人のクライアントが直接通信することを意味しません。クライアント間のすべての通信は、サーバーによって中継されます。

4. IRCプロトコルサービス

このセクションでは、IRCプロトコルが提供するサービスについて説明します。これらのサービスの組み合わせにより、リアルタイムの会議が可能になります。

4.1 クライアントロケーター

メッセージを交換できるようにするには、2人のクライアントがお互いを見つけることができなければなりません。

サーバーに接続すると、クライアントがラベルを使用して登録し、他のサーバーやクライアントがクライアントがどこにあるかを知るために使用されます。サーバーは、使用されているすべてのラベルを追跡する責任があります。

4.2 メッセージリレー

IRCプロトコルは、2人のクライアントが直接通信することを意味しません。クライアント間のすべての通信は、サーバーによって中継されます。

4.3 チャネルホスティングと管理

チャネルは、1人以上のユーザーの名前が付けられたグループであり、すべてがそのチャネルにアドレス指定されたメッセージを受信します。チャネルはその名前と現在のメンバーによって特徴付けられ、そのメンバーの一部によって操作できるプロパティのセットもあります。

チャネルは、複数のクライアントに送信されるメッセージに平均を提供します。サーバーはチャネルをホストし、必要なメッセージが多重化を提供します。サーバーは、チャネルメンバーを追跡することにより、チャネルの管理も担当します。サーバーの正確な役割は、「インターネットリレーチャット：チャネル管理」[IRC-chan]で定義されています。

5. IRCの概念

このセクションは、IRCプロトコルの構成の背後にある実際の概念と、異なるクラスのメッセージがどのように配信されるかを説明することに専念しています。

5.1 1対1のコミュニケーション

ほとんどのサーバーサーバートラフィックは、サーバーが互いに話し合う結果ではないため、通常、1対1の通信はクライアントによって実行されます。クライアントが互いに話し合う手段を提供するには、すべてのサーバーが、クライアントに到達す

message in exactly one direction along the spanning tree in order to reach any client. Thus the path of a message being delivered is the shortest path between any two points on the spanning tree.

The following examples all refer to Figure 1 above.

Example 1: A message between clients 1 and 2 is only seen by server A, which sends it straight to client 2.

Example 2: A message between clients 1 and 3 is seen by servers A & B, and client 3. No other clients or servers are allowed see the message.

Example 3: A message between clients 2 and 4 is seen by servers A, B, C & D and client 4 only.

5.2 One-To-Many

The main goal of IRC is to provide a forum which allows easy and efficient conferencing (one to many conversations). IRC offers several means to achieve this, each serving its own purpose.

5.2.1 To A Channel

In IRC the channel has a role equivalent to that of the multicast group; their existence is dynamic and the actual conversation carried out on a channel MUST only be sent to servers which are supporting users on a given channel. Moreover, the message SHALL only be sent once to every local link as each server is responsible to fan the original message to ensure that it will reach all the recipients.

The following examples all refer to Figure 2.

Example 4: Any channel with 1 client in it. Messages to the channel go to the server and then nowhere else.

Example 5: 2 clients in a channel. All messages traverse a path as if they were private messages between the two clients outside a channel.

Example 6: Clients 1, 2 and 3 in a channel. All messages to the channel are sent to all clients and only those servers which must be traversed by the message if it were a private message to a single client. If client 1 sends a message, it goes back to client 2 and then via server B to client 3.

5.2.2 To A Host/Server Mask

To provide with some mechanism to send messages to a large body of related users, host and server mask messages are available. These messages are sent to users whose host or server information match that of the mask. The messages are only sent to locations where users are, in a fashion similar to that of channels.

5.2.3 To A List

The least efficient style of one-to-many conversation is through clients talking to a 'list' of targets (client, channel, mask). How this is done is almost self explanatory: the client gives a list of destinations to which the message is to be delivered and the server breaks it up and dispatches a separate copy of the message to each given destination.

るためにスパニングツリーに沿って正確に一方向にメッセージを送信することが必要です。したがって、配信されるメッセージのパスは、スパニングツリー上の任意の2つのポイント間の最短パスです。

次の例はすべて、上記の図1を参照してください。

例1：クライアント1と2の間のメッセージは、サーバーAでのみ表示され、クライアント2に直接送信されます。

例2：クライアント1と3の間のメッセージは、サーバーA&Bとクライアント3によって表示されます。

例3：クライアント2と4の間のメッセージは、サーバーA、B、C & D、およびクライアント4のみで表示されます。

5.2 1対多い

IRCの主な目標は、簡単で効率的な会議（多くの会話）を可能にするフォーラムを提供することです。IRCは、これを達成するためのいくつかの手段を提供し、それぞれが独自の目的を果たしています。

5.2.1 チャンネルへ

IRCでは、チャンネルにはマルチキャストグループと同等の役割があります。それらの存在は動的であり、チャンネルで実行される実際の会話は、特定のチャンネルのユーザーをサポートしているサーバーにのみ送信する必要があります。さらに、各サーバーはすべての受信者に届くように元のメッセージをファンする責任があるため、メッセージはすべてのローカルリンクに1回しか送信されません。

次の例はすべて、図2を参照してください。

例4：1つのクライアントが入ったチャンネル。チャンネルへのメッセージはサーバーに移動し、他の場所には行きません。

例5：チャンネル内の2つのクライアント。すべてのメッセージは、チャンネルの外側の2つのクライアント間のプライベートメッセージであるかのようにパスを横断します。

例6：チャンネルのクライアント1、2、および3。チャンネルへのすべてのメッセージは、すべてのクライアントに送信され、単一のクライアントへのプライベートメッセージである場合、メッセージによって通過する必要があるサーバーのみが送信されます。クライアント1がメッセージを送信すると、クライアント2に戻り、サーバーBを介してクライアント3に戻ります。

5.2.2 ホスト/サーバーマスクに

関連するユーザーの大部分にメッセージを送信するメカニズムを提供するために、ホストとサーバーマスクのメッセージが利用可能です。これらのメッセージは、ホストまたはサーバー情報がマスクの情報と一致するユーザーに送信されます。メッセージは、チャンネルと同様の方法で、ユーザーがいる場所にも送信されます。

5.2.3 リストに

最も効率的でないスタイルの1対多様な会話は、クライアントがターゲット（クライアント、チャンネル、マスク）の「リスト」と話すことです。これがどのように行われるかはほとんど自明です。クライアントは、メッセージを配信するための目的地のリストを提供し、サーバーがそれを分割し、指定された各宛先にメッセージの個別のコピーを発送します。

This is not as efficient as using a channel since the destination list MAY be broken up and the dispatch sent without checking to make sure duplicates aren't sent down each path.

5.3 One-To-All

The one-to-all type of message is better described as a broadcast message, sent to all clients or servers or both. On a large network of users and servers, a single message can result in a lot of traffic being sent over the network in an effort to reach all of the desired destinations.

For some class of messages, there is no option but to broadcast it to all servers so that the state information held by each server is consistent between servers.

5.3.1 Client-to-Client

There is no class of message which, from a single message, results in a message being sent to every other client.

5.3.2 Client-to-Server

Most of the commands which result in a change of state information (such as channel membership, channel mode, user status, etc.) MUST be sent to all servers by default, and this distribution SHALL NOT be changed by the client.

5.3.3 Server-to-Server

While most messages between servers are distributed to all 'other' servers, this is only required for any message that affects a user, channel or server. Since these are the basic items found in IRC, nearly all messages originating from a server are broadcast to all other connected servers.

6. Current Problems

There are a number of recognized problems with this protocol, this section only addresses the problems related to the architecture of the protocol.

6.1 Scalability

It is widely recognized that this protocol does not scale sufficiently well when used in a large arena. The main problem comes from the requirement that all servers know about all other servers, clients and channels and that information regarding them be updated as soon as it changes.

6.2 Reliability

As the only network configuration allowed for IRC servers is that of a spanning tree, each link between two servers is an obvious and quite serious point of failure. This particular issue is addressed more in detail in "Internet Relay Chat: Server Protocol" [IRC-SERVER].

6.3 Network Congestion

Another problem related to the scalability and reliability issues, as well as the spanning tree architecture, is that the protocol and architecture for IRC are extremely vulnerable to network congestions. This problem is endemic, and should be solved for the next generation: if congestion and high traffic volume cause

これは、宛先リストが分割され、ディスパッチがチェックせずに送信され、複製が各パスに送信されないようにするため、チャンネルを使用するほど効率的ではありません。

5.3 1対1

すべてのタイプのメッセージは、すべてのクライアントまたはサーバー、またはその両方に送信されるブロードキャストメッセージとしてよりよく説明されています。ユーザーとサーバーの大規模なネットワークでは、単一のメッセージが、すべての目的の宛先に到達するために、ネットワーク上に多くのトラフィックが送信される可能性があります。

一部のクラスのメッセージについては、各サーバーが保持している状態情報がサーバー間で一貫しているように、すべてのサーバーにブロードキャストする以外にオプションはありません。

5.3.1 クライアントからクライアントへ

単一のメッセージから、他のすべてのクライアントに送信されるメッセージが表示されるメッセージのクラスはありません。

5.3.2 クライアントからサーバー

州情報の変更（チャンネルメンバーシップ、チャンネルモード、ユーザーステータスなど）の変更をもたらすコマンドのほとんどは、デフォルトですべてのサーバーに送信する必要があり、この分布はクライアントによって変更されません。

5.3.3 サーバーからサーバー

サーバー間のほとんどのメッセージはすべての「他の」サーバーに配布されますが、これはユーザー、チャンネル、またはサーバーに影響を与えるメッセージにのみ必要です。これらはIRCで見つかった基本的なアイテムであるため、サーバーから発信されたほぼすべてのメッセージが他のすべての接続されたサーバーにブロードキャストされます。

6. 現在の問題

このプロトコルには多くの認識された問題があります。このセクションでは、プロトコルのアーキテクチャに関連する問題のみに対処します。

6.1 スケーラビリティ

このプロトコルは、大きなアリーナで使用情况の場合、十分に十分に拡大しないことが広く認識されています。主な問題は、すべてのサーバーが他のすべてのサーバー、クライアント、およびチャンネルについて知っているという要件と、それらに関する情報が変更されるとすぐに更新されることから生じています。

6.2 信頼性

IRCサーバーに許可される唯一のネットワーク構成はスパニングツリーのネットワーク構成であるため、2つのサーバー間の各リンクは明らかで非常に深刻な障害点です。この特定の問題は、「インターネットリレーチャット：サーバープロトコル」[IRC-Server]で詳細に詳細に説明されています。

6.3 ネットワークの混雑

スケーラビリティと信頼性の問題、およびスパニングツリーアーキテクチャに関連する別の問題は、IRCのプロトコルとアーキテクチャがネットワーク輻輳に対して非常に脆弱であることです。この問題は風土病であり、次世代のために解決する必要があります。混雑と交通量の多いボリュームが2つのサーバー間の

a link between two servers to fail, not only this failure generates more network traffic, but the reconnection (eventually elsewhere) of two servers also generates more traffic.

In an attempt to minimize the impact of these problems, it is strongly RECOMMENDED that servers do not automatically try to reconnect too fast, in order to avoid aggravating the situation.

6.4 Privacy

Besides not scaling well, the fact that servers need to know all information about other entities, the issue of privacy is also a concern. This is in particular true for channels, as the related information is quite a lot more revealing than whether a user is online or not.

7. Security Considerations

Asides from the privacy concerns mentioned in section 6.4 (Privacy), security is believed to be irrelevant to this document.

8. Current Support And Availability

Mailing lists for IRC related discussion: General discussion: ircd-users@irc.org Protocol development: ircd-dev@irc.org

Software implementations: ftp://ftp.irc.org/irc/server ftp://ftp.funet.fi/pub/unix/irc ftp://coombs.anu.edu.au/pub/irc
--

Newsgroup: [alt.irc](#)

9. Acknowledgements

Parts of this document were copied from the RFC 1459 [IRC] which first formally documented the IRC Protocol. It has also benefited from many rounds of review and comments. In particular, the following people have made significant contributions to this document:

Matthew Green, Michael Neumayer, Volker Paulsen, Kurt Roeckx, Vesa Ruokonen, Magnus Tjernstrom, Stefan Zehl.

10. References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[IRC] Oikarinen, J. and D. Reed, "Internet Relay Chat Protocol", RFC 1459, May 1993.

[IRC-CLIENT] Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, April 2000.

[IRC-SERVER] Kalt, C., "Internet Relay Chat: Server Protocol", RFC 2813, April 2000.

[IRC-CHAN] Kalt, C., "Internet Relay Chat: Channel Management", RFC 2811, April 2000.

11. Author's Address

リンクを失敗させる場合、この障害がより多くのネットワークトラフィックを生成するだけでなく、2つのサーバーの再接続（最終的には他の場所）も生成しますより多くのトラフィック。

これらの問題の影響を最小限に抑えるために、状況を悪化させないように、サーバーが自動的に速く再接続しようとしなことを強くお勧めします。

6.4 プライバシー

よくスクリーニングしないことに加えて、サーバーが他のエンティティに関するすべての情報を知る必要があるという事実も、プライバシーの問題も懸念事項です。これは特にチャンネルに当てはまります。関連情報は、ユーザーがオンラインであるかどうかよりもはるかに明らかになっているためです。

7. セキュリティに関する考慮事項

セクション6.4（プライバシー）に記載されているプライバシーの懸念から、セキュリティはこのドキュメントとは無関係であると考えられています。

8. 現在のサポートと可用性

NewsGroup : [Alt.Irc](#)

9. 謝辞

このドキュメントの一部は、最初にIRCプロトコルを正式に文書化したRFC 1459 [IRC]からコピーされました。また、多くのラウンドのレビューとコメントの恩恵を受けています。特に、次の人々がこの文書に多大な貢献をしています。

マシュー・グリーン、マイケル・ノイマイヤー、ヴォルカー・ポールセン、カート・ロックズ、ヴェサ・ルーコネン、マグナス・ツェーンストロム、ステファン・ゼール。

10. 参考文献

[キーワード] Bradner, S. , 「要件レベルを示すためにRFCで使用するためのキーワード」、BCP 14、RFC 2119、1997年3月。

[IRC] Oikarinen、J。およびD. Reed、「インターネットリレーチャットプロトコル」、RFC 1459、1993年5月。

[IRC-Client] Kalt、C。 , 「インターネットリレーチャット：クライアントプロトコル」、RFC 2812、2000年4月。

[IRC-Server] Kalt、C。 , 「インターネットリレーチャット：サーバープロトコル」、RFC 2813、2000年4月。

[IRC-chan] Kalt、C。 , 「インターネットリレーチャット：チャンネル管理」、RFC 2811、2000年4月。

11. 著者の連絡先

EMail: kalt@stealth.net

12. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

12. 完全な著作権声明

Copyright (c) The Internet Society (2000)。全著作権所有。

このドキュメントと翻訳は他の人にコピーされて提供される場合があり、それについてコメントまたは説明するか、その実装を支援する派生作品は、いかなる種類の制限なしに、準備、コピー、公開、配布される場合があります。、上記の著作権通知とこの段落がそのようなすべてのコピーとデリバティブ作品に含まれている場合。ただし、このドキュメント自体は、インターネット協会や他のインターネット組織への著作権通知や参照を削除するなど、いかなる方法でも変更できない場合があります。インターネット標準のプロセスに従うか、英語以外の言語に翻訳するために必要な場合に従う必要があります。

上記の限られた許可は永続的であり、インターネット社会またはその後継者または譲受人によって取り消されることはありません。

この文書と本書に含まれる情報は、「現状」に基づいて提供されており、インターネット社会とインターネットエンジニアリングタスクフォースは、ここにある情報の使用が行われないという保証を含むがこれらに限定されないすべての保証を否認します。特定の目的に対する商品性または適合性の権利または黙示的な保証を侵害します。

謝辞

RFCエディター機能の資金は現在、インターネット協会によって提供されています。