

# RFC 2810

## Introduction

IRC 프로토콜은 텍스트 기반 회의와 함께 사용하도록 수년에 걸쳐 설계되었습니다. 이 문서는 현재 아키텍처를 설명합니다.

IRC 프로토콜은 클라이언트-서버 모델을 기반으로 하며 분산 된 방식으로 많은 컴퓨터에서 실행하는 데 적합합니다. 일반적인 설정에는 클라이언트(또는 다른 서버)가 연결할 중앙 지점을 형성하고 필요한 메시지 전달/멀티플렉싱 및 기타 기능을 수행하는 단일 프로세스(서버)가 포함됩니다.

각 서버가 전역 상태 정보의 복사본을 가져야하는 이 분산 모델은 네트워크가 도달할 수 있는 최대 크기를 제한하는 심각한 핸디캡이므로 프로토콜의 가장 심각한 문제입니다. 기존 네트워크가 놀라운 속도로 계속 성장할 수 있었다면 더욱 강력한 시스템을 제공 한 하드웨어 제조업체에 감사해야 합니다.

## Components

다음 단락은 IRC 프로토콜의 기본 구성 요소를 정의합니다.

### Server

서버는 다른 모든 구성 요소를 함께 연결할 수 있는 프로토콜의 유일한 구성 요소이므로 IRC의 중심을 형성합니다. 클라이언트가 연결하여 서로 통신 할 수 있는 지점[IRC\_CLIENT]과 지점을 제공합니다. 다른 서버가 [IRC\_SERVER]에 연결할 수 있습니다. 서버는 또한 IRC 프로토콜에 정의 된 기본 서비스를 제공 할 책임이 있습니다.

### Clients

클라이언트는 다른 서버가 아닌 서버에 연결하는 모든 것입니다. 서로 다른 목적을 제공하는 두 가지 유형의 클라이언트가 있습니다.

- User Client
  - 일반적으로 IRC를 통해 대화식으로 통신하는 데 사용되는 텍스트 기반 인터페이스를 제공하는 프로그램
  - User라고도 함
- Service Client

- 수동으로 사용하거나 대화를 위해 사용할 수 없음
- 프로토콜의 채팅 기능에 대한 더 제한된 액세스 권한을 가짐
- 선택적으로 서버에서 더 많은 개인 데이터에 액세스 할 수 있음
- 서비스는 일반적으로 사용자에게 일종의 서비스(IRC 자체와 관련이 없는)를 제공하는 데 사용되는 자동 장치
- IRC 네트워크에 연결된 사용자의 출처에 대한 통계를 수집하는 서비스 등등

## Architecture

IRC 네트워크는 서로 연결된 서버 그룹에 의해 정의됩니다. 단일 서버는 가장 단순한 IRC 네트워크입니다.

IRC 서버에 허용되는 유일한 네트워크 구성은 각 서버가 보고 있는 나머지 네트워크의 중앙 노드 역할을 하는 스페닝 트리의 구성입니다. IRC 프로토콜은 두 클라이언트가 직접 통신 할 수 있는 수단을 제공하지 않습니다. 클라이언트 간의 통신은 서버(들)에 의해 중계됩니다.

## IRC Protocol Services

이 섹션에서는 IRC 프로토콜에서 제공하는 서비스에 대해 설명합니다. 이러한 서비스의 조합을 통해 실시간 회의가 가능합니다.

### Client Locator

메시지를 교환하려면 두 클라이언트가 서로를 찾을 수 있어야 합니다.

서버에 연결할 때 클라이언트는 레이블을 사용하여 등록합니다. 이 레이블은 다른 서버와 클라이언트가 클라이언트 위치를 알 수 있도록 사용됩니다. 서버는 사용중인 모든 레이블을 추적 할 책임이 있습니다.

### Message Relaying

IRC 프로토콜은 두 클라이언트가 직접 통신 할 수 있는 수단을 제공하지 않습니다. 클라이언트 간의 모든 통신은 서버에 의해 중계됩니다.

### Channel Hosting And Management

채널은 해당 채널로 주소가 지정된 메시지를 모두 수신하는 한 명 이상의 사용자로 구성된 이름 있는 그룹입니다. 채널은 이름과 현재 회원으로 특징지어지며, 회원이 조작할 수 있는 속성 세트도 있습니다.

채널은 여러 클라이언트에게 메시지를 보낼 수 있는 수단을 제공합니다. 서버는 채널을 호스팅하여 필요한 메시지 멀티플렉싱을 제공합니다. 서버는 또한 채널 구성원을 추적하여 채널을 관리 할 책임이 있습니다. 서버의 정확한 역할은 “RFC 2811”을 참고하면 됩니다.

## IRC Concepts

이 섹션은 IRC 프로토콜의 구성 뒤에 있는 실제 개념과 다양한 메시지 클래스가 전달되는 방법을 설명하는 데 전념합니다.

### One-To-One Comunication

대부분의 서버-서버 트래픽은 서버가 서로만 통신하는 결과가 아니기 때문에 일대일 통신은 일반적으로 클라이언트에 의해 수행됩니다. 클라이언트가 서로 통신 할 수 있는 수단을 제공하려면 모든 서버가 클라이언트에 도달하기 위해 스페닝 트리를 따라 정확히 한 방향으로 메시지를 보낼 수 있어야 합니다. 따라서 전달되는 메시지의 경로는 스페닝 트리의 두 지점 사이의 최단 경로입니다.

### One-To-Many

IRC의 주요 목표는 쉽고 효율적인 회의(일대 다 대화)가 가능한 포럼을 제공하는 것입니다. IRC는 이를 달성하기 위한 여러 가지 방법을 제공하며, 각 방법은 고유 한 목적에 부합됩니다.

- To a Channel
  - 채널은 멀티 캐스트 그룹과 동일한 역할
  - 동적으로 존재하며 채널에서 수행되는 실제 대화는 주어진 채널에서 사용자를 지원하는 서버로만 전송되어야 함
  - 메시지는 각 서버가 모든 수신자에게 도달 할 수 있도록 원본 메시지를 부채질 할 책임이 있으므로 모든 로컬 링크에 한 번만 전송됩니다.
- To a Host/Server Mask
  - 다수의 관련 사용자에게 메시지를 보내는 메커니즘을 제공하기 위해 호스트 및 서버 마스크 메시지를 사용할 수 있음
  - 이러한 메시지는 호스트 또는 서버 정보가 마스크의 정보와 일치하는 사용자에게 전송됨
  - 메시지는 채널과 유사한 방식으로 사용자가 있는 위치로만 전송됨
- To A List

- 일대 다 대화의 가장 효율적인 스타일은 클라이언트가 대상(클라이언트, 채널, 마스크)의 리스트와 대화하는 것
- 클라이언트는 메시지가 전달 될 대상 리스트를 제공하고 서버는 메시지를 분리하고 지정된 각 대상에 별도의 메시지 복사본을 발송
- 대상 리스트가 분리 될 수 있고 중복 항목이 각 경로로 전송되지 않는지 확인하지 않고 발송이 전송 될 수 있으므로 이는 채널을 사용하는 것만큼 효율적이진 않음

## One-To-All

one-to-all 유형의 메시지는 모든 클라이언트나 서버 또는 둘 다에 전송되는 브로드 캐스트 메시지로 더 잘 설명됩니다. 대규모 사용자 및 서버 네트워크에서 단일 메시지로 인해 원하는 모든 대상에 도달하기 위해 네트워크를 통해 많은 트래픽이 전송 될 수 있습니다.

일부 메시지 클래스의 경우 각 서버가 보유한 상태 정보가 서버간에 일관되도록 모든 서버에 브로드 캐스트 하는 것 외에는 옵션이 없습니다.

- Client-to-Client
  - 단일 메시지에서 다른 모든 클라이언트로 메시지가 전송되는 메시지 클래스는 없음
- Client-to-Server
  - 상태 정보(채널 멤버십, 채널 모드, 사용자 상태 등)를 변경하는 대부분의 명령은 기본적으로 모든 서버로 전송되어야 함
  - 이 배포는 클라이언트에 의해 변경되지 않아야 함
- Server-to-Server
  - 서버 간의 대부분의 메시지는 모든 '기타'서버로 분산되지만 사용자, 채널 또는 서버에 영향을 미치는 메시지에만 필요함
  - 이들은 IRC에서 발견되는 기본 항목이므로 서버에서 발생하는 거의 모든 메시지가 연결된 다른 모든 서버로 브로드 캐스트됨

## Current Problems

이 프로토콜에는 알려진 여러 문제가 있으며 이 섹션에서는 프로토콜의 아키텍처와 관련된 문제만 다룹니다.

## Scalability

이 프로토콜은 대규모 경기장에서 사용할 때 충분히 확장되지 않는다는 것이 널리 알려져 있습니다. 주요 문제는 모든 서버가 다른 모든 서버, 클라이언트 및 채널에 대해 알고 있고 이에 관한 정보가 변경되는 즉시 업데이트 된다는 요구 사항에서 비롯됩니다.

## Reliability

IRC 서버에 허용되는 유일한 네트워크 구성은 스페닝 트리 구성이므로 두 서버 간의 각 링크는 분명하고 매우 심각한 실패 지점입니다. 이 특정 문제는 “RFC 2813”에서 자세히 설명합니다.

## Network Congestion

확장성 및 안정성 문제와 스페닝 트리 아키텍처와 관련된 또 다른 문제는 IRC의 프로토콜 및 아키텍처가 네트워크 정체에 매우 취약하다는 것입니다. 이 문제는 고유하며 다음 세대를 위해 해결해야 합니다. 정체와 높은 트래픽 볼륨으로 인해 두 서버 간의 링크가 실패하면 이 실패는 더 많은 네트워크 트래픽을 생성할 뿐만 아니라 두 서버의 재연결(다른 곳)에서도 더 많은 트래픽을 생성합니다.

이러한 문제의 영향을 최소화 하기 위해, 상황이 악화되지 않도록 서버가 자동으로 너무 빨리 재연결을 시도하지 않도록 하는 것이 좋습니다.

## Privacy

잘 확장되지 않는 것 외에도 서버가 다른 객체에 대한 모든 정보를 알아야 한다는 사실로 인해 프라이버시 문제가 생깁니다. 사용자가 온라인 상태인지 아닌지보다 관련 정보가 훨씬 더 많이 드러나기 때문에 채널의 경우는 특히 그렇습니다.

## Security Considerations

위의 개인 정보 보호 문제를 제외하고 보안은 이 문서와 관련이 없는 것으로 간주됩니다.

## References

IRC: RFC 1459 IRC-CLIENT: RFC 2812 IRC-SERVER: RFC 2813 IRC-CHANNEL: RFC 2811

## Full Copyright Statements

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.