

INTERPRETANDO EL RFC DE IRC (INTERNET RELAY CHAT)

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

30 de enero de 2013

Index

- 1 Introduction
- 2 Especificaciones de IRC
- 3 Conceptos de IRC
- 4 Mensajes
- 5 Respuestas
- 6 Problemas

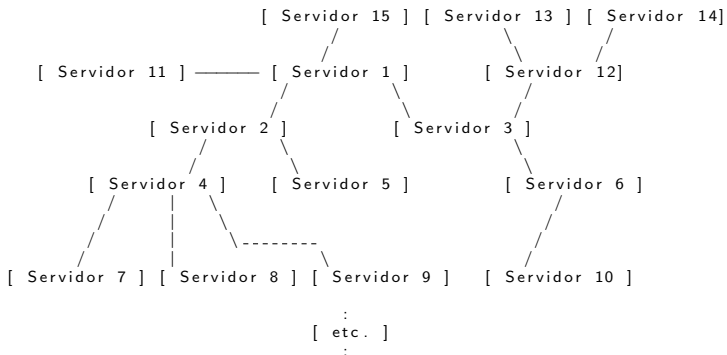
Introducción

- El protocolo IRC se ha desarrollado en sistemas que usan el protocolo de red TCP/IP, aunque no es imperativo que esta sea la única forma en que funcione.
- Una configuración típica incluye un único proceso (el servidor) que conforma un punto central para que los clientes (u otros servidores) se conecten a él, realizando los envíos y multiplexado de mensajes requeridos, así como otras funciones.

Servidores

- Forman la espina dorsal del IRC.
- Proporcionan un punto al que los clientes pueden conectar para hablar unos con otros, y un punto para que otros servidores se conecten a él, formando una red IRC.
- La única configuración de red permitida para los servidores es un árbol extendido.
- Cada servidor hace de nodo central para el resto de la red que dicho servidor ve.

Servidores



Clientes

- Un cliente es cualquier cosa que se conecta a un servidor que no sea otro servidor.
- Se distingue de otros clientes por un único nick de 9 caracteres de longitud máxima.
- todos los servidores deben tener la siguiente información sobre todos los clientes:
 - Nombre real del host desde el que conecta el cliente.
 - El nombre de usuario del cliente en ese host.
 - El servidor al que está conectado el cliente.

Operadores

- Clientes que se utilizan para mantener un cierto orden en la red de IRC.
- Deben ser capaces de realizar tareas básicas de red como desconectar y reconectar servidores para prevenir un uso prolongado de mal rutado de red.
- Pueden eliminar un usuario de la red por la “fuerza”.

Canales

- Es un grupo.
- Se crea implícitamente al unirse el primer cliente, y deja de existir cuando el último cliente lo deja.
- Los nombres de canales son cadenas. La única restricción del nombre es que no puede contener espacios en blanco (' '), control G, o una coma (',').
- Hay dos tipos de canales:
 - Uno es un canal distribuido que es conocido por todos los servidores de la red. Estos canales se marcan con el primer carácter '#'.
 - Otro tipo de canales se caracteriza por ser sólo para clientes conectados al servidor en el que se encuentra el canal. Se distinguen porque su primer carácter es '&'.

Canales

- Para crear un nuevo canal o formar parte de uno existente, un usuario debe UNIRSE (JOIN) al canal.
- Si el canal no existe, se crea y el creador del canal pasa a ser operador de canal.
- Un usuario puede formar parte de varios canales simultáneamente, pero se recomienda un límite de 10.

Operadores de Canales

- Se le considera el “dueño” de dicho canal.
- Poseen ciertos “poderes” que les permiten mantener el control y cierto orden en su canal.
- No tiene que dar justificaciones por sus actos.
- Se identifica por el símbolo '@’.

Operadores de Canales

- Utilizan los siguientes comandos:
 - KICK - Expulsar a un cliente del canal, de forma que puede volver a entrar
 - MODE - Cambiar los modos del canal
 - INVITE - Invitar a un usuario a un canal
 - TOPIC - Cambiar el topic en un canal con modo +t

Discusión

- Hay dos tipos de conexiones principales:
 - Servidor-servidor.
 - Cliente-servidor.

Mensajes

- Servidores y clientes se envían mensajes unos a otros.
- Si el mensaje contiene un comando válido se suele esperar una respuesta.
- La comunicación cliente-servidor y servidor-servidor es esencialmente asíncrona.
- Cada mensaje consta de 3 partes principales (separadas por espacios):
 - El prefijo (opcional, su presencia se indica con ':' al inicio).
 - El comando.
 - Los parámetros del comando (hasta un total de 15).

Mensajes: El prefijo

- Lo usan los servidores para indicar el verdadero origen del mensaje.
- Los clientes no deberían usar prefijos al enviar mensajes.
- El único válido es el nick registrado asociado con el cliente.

Mensajes

- Los mensajes de IRC siempre son líneas de caracteres acabadas en un par CR-LF.
- No deben exceder los 512 caracteres de longitud, incluido el par CR-LF.

Formato de mensajes en 'pseudo' BNF

- CR y LF son separadores de mensajes.
- Los mensajes vacíos se ignoran de forma silenciosa.
- El mensaje extraído se divide en las componentes <prefijo>, <comando> y lista de parámetros formada por componentes <parámetro intermedio> o <parámetro final>.

Formato de mensajes en 'pseudo' BNF

```

<mensaje> ::= [ ':' <prefijo> <ESPACIO> ] <comando> <parametro> <CrLf>
<prefijo>  ::= <nombre de servidor> | <nick> [ '!' <usuario> ] [ '@' <host> ]
<comando> ::= <letra> { <letra> } | <numero> <numero> <numero>
<ESPACIO> ::= ' ' { ' ' }
<parametro> ::= <ESPACIO> [ ':' <parametro final> |
                               <parametro intermedio> <parametro> ]

<parametro intermedio> ::= <Cualquier secuencia de octetos *no vacia*
                           que no incluya ESPACIO, NUL, CR o LF, el
                           primero del cual no puede ser ':'>
<parametro final> ::= <Cualquier secuencia, posiblemente *vacía* que no
                       incluya NUL, CR o LF>

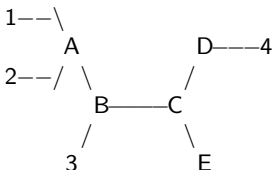
<CrLf> ::= CR LF
  
```

Respuestas numéricas

- La mayoría de los mensajes enviados al servidor generan una respuesta de alguna clase.
- La respuesta más común es la numérica, empleada tanto para repuestas de error como para las normales.

Conceptos de IRC

- Para explicar las comunicaciones usaremos el siguiente árbol.



Servidores: A, B, C, D, E

Clientes: 1, 2, 3, 4

Comunicación uno-a-uno

- Sólo la realizan los clientes.
- El camino de un mensaje es el más corto entre dos puntos cualesquiera del árbol.

Comunicación Uno-a-muchos

- El propósito principal del IRC es proporcionar un forum que permita realizar conferencias de forma sencilla y eficiente.

A una lista

- La forma menos eficiente.
- El cliente da una lista de destinos para un mensaje y el servidor divide la lista y distribuye una copia separada del mensaje a cada destino.

A un grupo (canal)

- Su existencia es dinámica
- La conversación se envía únicamente a los servidores que tienen usuarios en el canal.

A una máscara de host o servidor

- Para proveer a los Operadores de IRC de mecanismos para enviar mensajes a muchos usuarios relacionados
- Estos mensajes se envían a usuarios cuya información de host o servidor coincide con la de la máscara.

Uno-a-todos

- Mensaje de emisión, enviado a todos los clientes, servidores o ambos (broadcast)

Cliente-a-cliente

- No existe

Cliente-a-servidor

- La mayoría de los comandos que resultan en un cambio en la información sobre el estado

Servidor-a-servidor

- Casi todos los mensajes originados de un servidor se envían a todos los demás servidores conectados

Mensajes

- Formato más usual:

```
:Nombre COMANDO lista de parámetros
```

Registro de conexión

- 1 Mensaje de Password
- 2 Mensaje de Nick
- 3 Mensaje de Usuario

Mensaje de Password

- Comando: PASS
- Parámetros: <password>
- Descripción: para establecer una clave de conexión
- Ejemplo: PASS clavesecretaaquí

Mensaje de Nick

- Comando: NICK
- Parámetros: <nick> [<contadordesalto>]
- Descripción: para darle al usuario un nick o cambiar el anterior
- Ejemplo: NICK Wiz ; Introduciendo nuevo nick "Wiz".

Mensaje de Usuario

- Comando: USER
- Parámetros: <nombre de usuario> <nombre de host>
<nombre de servidor> <nombre real>
- Descripción: al principio de cada conexión para indicar el nombre de usuario, de host y servidor y el nombre real del nuevo usuario
- Ejemplo: USER guest tolmoon tolsun :Ronnie Reagan
;El usuario se registra con nombre de usuario "guest" y nombre real Ronnie Reagan".

Mensaje de Servidor

- Comando: SERVER
- Parámetros: <nombre de servidor> <contador de salto>
<información>
- Descripción: para indicar a un servidor que el otro lado de la conexión es un servidor
- Ejemplo: `SERVER test.oulu.fi 1 :[tolsun.oulu.fi]`
Servidor experimental ; El nuevo servidor test.oulu.fi se presenta e intenta registrarse. El nombre entre corchetes es el nombre de host que lleva test.oulu.fi.

Oper

- Comando: OPER
- Parámetros: <usuario> <password>
- Descripción: para que un usuario normal obtenga privilegios de Operador
- Ejemplo: OPER foo bar ; Intenta registrarse como Operador usando el nombre de usuario “foo” y la clave “bar”

Mensaje de salida

- Comando: QUIT
- Parámetros: [<Mensaje de salida>]
- Descripción: Una sesión de un cliente se finaliza con un mensaje de salida.
- Ejemplo: QUIT :Me voy a comer ; Formato de mensaje

Mensaje de salida del servidor

- Comando: SQUIT
- Parámetros: <servidor> <comentario>
- Descripción: . Si un servidor quiere terminar la conexión con otro servidor, debe enviar un mensaje SQUIT al otro servidor, con el nombre del otro servidor como parámetro
- Ejemplo: SQUIT tolsun.oulu.fi ;¿Enlace erróneo? ;El enlace del servidor tolsun.oulu.fi ha finalizado por “Enlace erróneo”

Mensajes

- Se refiere a la manipulación de canales, sus propiedades (modos de canal) y sus contenidos (normalmente clientes)

Mensajes

- Comando: JOIN
- Parámetros: <canal>{,<canal>} [<clave>{,<clave>}]
- Descripción: para empezar a escuchar un canal específico.
- Ejemplo: JOIN # foobar ; unirse al canal # foobar.

Mensaje de salida del canal

- Comando: PART
- Parámetros: [<Mensaje de salida>]
- Descripción: provoca el borrado de la lista de usuarios activos de todos los canales dados en la lista de parámetros.
- Ejemplo: PART # twilightzone ; abandonar el canal “# twilightzone”

Mensaje de modos

- Comando: MODE
- Parámetros: <canal> { [+|-]| o| p| s| i| t| n| b| v }
[<límite>] [<usuario>] [<máscara de ban>] | <nick> { [+|-]| i| w| s| so }
- Descripción: Permite cambiar los modos tanto a los usuarios como a los canales.
- Ejemplo1: MODE # Finnish +im ; El canal # Finnish es ahora moderado y sólo para invitados
- Ejemplo2: :Angel MODE Angel +i ; Mensaje de Angel para hacerse invisible

Mensaje de tópico

- Comando: TOPIC
- Parámetros: <canal> [<tópico>]
- Descripción: para cambiar o ver el tópico de un canal.
- Ejemplo: `:Wiz TOPIC # test :Nuevo tópico ;`El usuario WiZ pone un tópico

Mensaje de nombres

- Comando: NAMES
- Parámetros: [`<canal>`{`,<canal>`}]
- Descripción: listar todos los nicks que sean visibles en cualquier canal que pueda ver.
- Ejemplo: `NAMES # twilightzone,# 42` ;listar usuarios visibles en `# 42` y `# twilightzone` si puedes ver los canales

Mensaje de lista de canales

- Comando: LIST
- Parámetros: [`<canal>`{`,<canal>`} [`<servidor>`]]
- Descripción: para listar los canales y sus tópicos.
- Ejemplo: LIST # twilightzone, # 42 ;Listar canales # twilightzone y # 42

Mensaje de invitación a un canal

- Comando: INVITE
- Parámetros: <nick> <canal>
- Descripción: para invitar a otros usuarios a un canal.
- Ejemplo: `:Angel INVITE Wiz # Dust ;Angel invita a WiZ al canal # Dust`

Comando de expulsión temporal

- Comando: KICK
- Parámetros: <canal> <usuario> [<comentario>]
- Descripción: para eliminar a un usuario de la lista de miembros de un canal
- Ejemplo: KICK & Melbourne Matthew ; Expulsar a Matthew de & Melbourne

Peticiones y comandos del servidor

- Para devolver información de cualquier servidor conectado a la red.
- Mirar RFC

Enviando mensajes

- Para la comunicación entre clientes.
- PRIVMSG y NOTICE son los únicos comandos disponibles que envían mensajes de texto de un cliente a otro

Mensajes privados

- Comando: PRIVMSG
- Parámetros: <receptor>{,<receptor>} <texto>
- Descripción: para enviar mensajes privados o públicos entre usuarios
- Ejemplos:
 - :Angel PRIVMSG Wiz :Hola ¿recibes esto? ; Mensaje de Angel a Wiz.
 - PRIVMSG Angel :Sí que lo recibo :) ;Mensaje para Angel.
 - PRIVMSG # Room :Buenas ;Mensaje para la sala #Room.
 - PRIVMSG jto@tolsun oulu.fi :Hola ; Mensaje a un cliente en el servidor tolsun oulu.fi con nombre de usuario "jto"
 - PRIVMSG \$*.fi :Servidor tolsun oulu.fi reiniciando ; Mensaje a todos los conectados a un servidor de máscara *.fi

Mensajes de aviso

- Comando: NOTICE
- Parámetros: <nick> <texto>
- Descripción: se usa de forma similar a PRIVMSG. La diferencia entre ambos es que no se pueden enviar respuestas automáticas a un NOTICE

Peticiones de usuario

- Comandos relacionados con la búsqueda de detalles sobre un usuario o grupo de usuarios en concreto
- Mirar RFC

Otros mensajes

- No categorizados

Mensajes

- Comando: Mensaje de “KILL”
- Parámetros: <nick> <comentario>
- Descripción: para cerrar una conexión cliente-servidor, mediante el servidor que sostiene la conexión
- Ejemplo: KILL David (csd.bu.edu <- tolsun.oulu.fi)
;Colisión de nicks entre csd.bu.edu y tolsun.oulu.fi

Mensaje de “PING”

- Comando: PING
- Parámetros: <servidor1> [<servidor2>]
- Descripción: para comprobar la presencia de un cliente activo al otro lado de la conexión
- Ejemplo: PING WiZ ;mensaje PING enviado al nick WiZ

Mensaje de “PONG”

- Comando: PONG
- Parámetros: <demonio> [<demonio2>]
- Descripción: El mensaje de PONG es la respuesta al mensaje de PING
- Ejemplo: PONG csd.bu.edu tolsun.oulu.fi ;mensaje PONG desde csd.bu.edu a tolsun.oulu.fi

Error

- Comando: ERROR
- Parámetros: <mensaje de error>
- Descripción: El comando ERROR lo usan los servidores para informar sobre errores serios o fatales a sus Operadores
- Ejemplo: ERROR :El servidor *.fi ya existe ;mensaje de ERROR al servidor que causó el error

Respuestas

- Hay 3 tipos:
 - Error.
 - Respuestas a comandos.
 - Reservadas
- **EJERCICIO: Ver las posibles respuestas en las 3 categorías**

Problemas

- Escalabilidad: no escala suficientemente bien cuando se usa de una forma muy masiva
- Etiquetas: colisiones.
- Algoritmos: no se han podido evitar algoritmos $O(N^2)$