

Introduction to IRC via irssi

Wade Cline

September 10, 2019

Outline

About

Topology

Interface

Chatting

Modes

Services

Netsplits

SSL/TLS

Questions

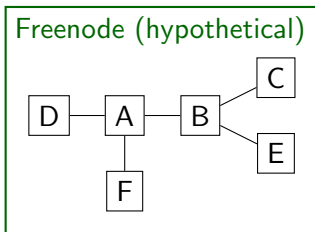
- 1 About
- 2 Topology
- 3 Interface
- 4 Chatting
- 5 Modes
- 6 Services
- 7 Netsplits
- 8 SSL/TLS
- 9 Questions

Examples in `irssi`

- Internet Relay Chat (IRC)
- Real-time, text-based chat for groups
 - Open specifications
 - RFC 1459 (Internet Relay Chat Protocol)
 - RFC 2810 (Internet Relay Chat: Architecture)
 - RFC 2811 (Internet Relay Chat: Channel Management)
 - RFC 2812 (Internet Relay Chat: Client Protocol)
 - RFC 2813 (Internet Relay Chat: Server Protocol)
 - Free Software implementations
 - Clients: `irssi`, `weechat`, `HexChat`, `KiwiIRC`, &c.
 - Servers: `InspIRCd`, `ircd-seven`, `UnrealIRCd`, `Charybdis`, &c.
 - Stable protocol
 - Minimal

Topology

- Each IRC service is a separate *network* (Ex: Freenode, OFTC, PDXCAT)
- A network consists of one or more *servers* (Ex: `orwell.freenode.net`, `card.freenode.net`, `barjavel.freenode.net`)
- Servers arranged in a *spanning tree* topology. Hypothetical Freenode network with servers joined in order A - F:



Connecting

- *Clients* (such as `irssi`) connect to individual servers, and thus join the network.
- Begin by running `irssi`.
- Configure the Freenode network: `/network add freenode`.
- Add a server to the network: `/server add -network freenode chat.freenode.net`
- Connect to the network: `/connect freenode`
- Server will display its Message of the Day (MotD)
- Auto-connect on startup via `/server add -auto chat.freenode.net`
- Close with `/exit`, then rerun via `irssi`.

irssi interface

About

Topology

Interface

Chatting

Modes

Services

Netsplits

SSL/TLS

Questions

```

1 Irssi v1.2.2 - https://irssi.org
    sustained support in keeping the
17:15 -!- - network going for close to two decades.
17:15 -!- -
17:15 -!- - Thank you to all of our attendees, sponsors,
2 speakers, exhibitors, helpers,
17:15 -!- - and everyone else who made this year's freenode
  #live conference amazing.
17:15 -!- - https://freenode.net/news/live-2018
17:15 -!- -
17:15 -!- - Thank you for using freenode!
17:15 -!- End of /MOTD command.
17:15 -!- Mode change [+i] for user frostsno1
17:15 -freenode-connect(frigg@freenode/utility-bot/frigg)-
    Welcome to freenode. To protect the network all new
    connections will be scanned for vulnerabilities.
    This will not harm your computer, and vulnerable
    hosts will be notified.
17:15 freenode-connect [frigg@freenode/utility-bot/frigg]
3 requested CTCP VERSION from frostsno1:
4 17:15 -!- Irssi: Your nick is owned by Unknown
5 [~frostsnow@static-50-43-47-183.bvtn.or.frontiernet.net]
6 [17:23] [frostsno1(+i)] [1:freenode (change with [Act: 2,3,4]
7 [(status)] ]

```

irssi interface (cont.)

- ① Client version
- ② Window text
- ③ Current time
- ④ Nickname and user modes
- ⑤ Window number and current network.
- ⑥ Window numbers containing **activity**.
 - Light blue: Informational messages
 - White: Chat messages
 - Purple: Messaging *you*
- ⑦ Input bar

- Person running client known as the *user*.
- *User* identified in IRC by *nickname*, or "nick"
 - Also known as *handle*
 - The most important identifier
 - Set via `/nick <name>`
- Additional *ident* entry is username on user's machine.
- Last, *realname* is user's actual name.

User info

About

Topology

Interface

Chatting

Modes

Services

Netsplits

SSL/TLS

Questions

- `/whois <nick>` (and `/whowas`) to view user information

```
18:21 -!- frostsno1
[~frostsnow@static-50-43-47-183.bvtn.or.frontiernet.net]
18:21 -!- ircname : Unknown
18:21 -!- channels : #gentoo
18:21 -!- server : barjavel.freenode.net [Paris, FR, EU]
18:21 -!- hostname :
static-50-43-47-183.bvtn.or.frontiernet.net
50.43.47.183
18:21 -!- idle : 0 days 1 hours 6 mins 56 secs [signon:
Mon Sep 2 17:15:03 2019]
18:21 -!- End of WHOIS
```

- Topmost: Nickname [ident@hostmask]
 - Hostmask identifies users, often obfuscated by default (but not on Freenode)
- ircname: User's real name (settings - core - real_name in .irssi/config)
- Channels user is joined to (not always displayed)
- Other information (accounts, cert fingerprint, &c.)

- Groups of users communicate over *channels*
- Channel *names* are prefixed by a #
 - ## prefix for "off-topic" channels in Freenode
- Enter a channel by *joining* it (/join <name>)
 - Ex: /join #osu-lug
- Channel *topic* provides information (/topic)
- View user list with /names
- Leave a channel by *parting* it (/part)

```

Oregon State University Linux Users Group | Tuesdays 18:00 KEC 1007 | http:
1 [~frostsnow@static-50-43-47-183.bvtn.or.frontiernet.net] has
  joined #osu-lug
20:16 -!- Topic for #osu-lug: Oregon State University Linux Users Group |
  Tuesdays 18:00 KEC 1007 | http://lug.oregonstate.edu | Freenode
  policies apply: https://freenode.net/news/the-new-policies
20:16 -!- Topic set by edunham [] [Tue Nov 7 17:17:10 2017]
20:16 [Users #osu-lug]
20:16 [+magical] [edunham] [Marajade] [resistor1]
20:16 [+pop] [eldebrim] [Mathuin] [runnerboi]
20:16 [Ac-town] [finchd] [murrown] [sl424]
20:16 [armoredPotato] [frostsn01] [Odysimus] [sorbet]
20:16 [bkero] [frostsnow] [pono] [thallia]
20:16 [bramwelt] [habit_] [pumpkin87] [thox]
20:16 [bryon] [localhost00] [penguin] [Vancelot]
20:16 [catherin1] [lrr] [Ramereth] [voxelv[m]]
20:16 [chancez] [lucyw] [recyclops]
20:16 [doublej472] [magica] [relud_]
20:16 -!- Irssi: #osu-lug: Total of 38 nicks [0 ops, 0 halfops, 2 voices,
  36 normal]
20:16 -!- Channel #osu-lug created Sat Nov 25 22:42:43 2006
20:16 -!- Irssi: Join to #osu-lug was success in 2 secs
[21:06] [frostsn01(+)] [2:freenode/#osu-lug(+nt)] [Act: 3]
[#osu-lug]

```

- ① Channel topic (/topic to view long topics)
- ② Channel users (/names if you forget)
- ③ Window number:network/channel name(channel modes)
- ④ Channel

```
<@frostsno1> Saying something  
* frostsno1 doing something  
<@frostsno1> nobody: ping  
<@frostsno1> bold *bold*  
<@frostsno1> underline  
<@frostsno1> green
```

- 1 Type words for raw text
- 2 `/me` command for actions (emotes)
- 3 *pinging* a user (tab-complete their name)
- 4 Bold with Ctrl+B or asterisks
- 5 Underline with underscores
- 6 Ctrl+C+# for mIRC color codes
 - Codes: `/usr/share/doc/irssi/formats.txt`

- Change windows with `/window #` (or `Alt+#`)
- Auto-join channel: `/channel add -auto <channel> <network>`
 - Ex: `/channel add -auto #osu-lug freenode`
- List network channels with `/list`
 - Warning: Don't use on Freenode due to channel count
- Direct message a nick via `/msg <nick>`
 - Also known as a *Private Message* (PM)
 - Generally considered rude unless invited
- History not saved by default
 - Save locally with: `/set autolog on`

- Basically configuration settings
- Represented by a single, case-sensitive letter
- Two kinds: *user modes* and *channel modes*
- Certain modes are *parameterized*
- Mode list may vary between networks
- `/help mode` vs `/quote help umode` and `/quote help cmode`
- Many are configurable via *services*

User mode examples

[About](#)[Topology](#)[Interface](#)[Chatting](#)[Modes](#)[Services](#)[Netsplits](#)[SSL/TLS](#)[Questions](#)

- **i: invisible**
 - User not listed in `/who` unless "known"
- **R: Registered DMs only (Freenode-specific)**
 - Only registered users may direct-message the user
- **Z: Secure connection (Freenode-specific)**
 - User connected via SSL/TLS
- **o: IRC Operator (IRCop)**
 - User is an IRC operator

Channel mode examples

[About](#)[Topology](#)[Interface](#)[Chatting](#)[Modes](#)[Services](#)[Netsplits](#)[SSL/TLS](#)[Questions](#)

- **t**: Topic lock
 - Only channel operators may set channel topic
- **s**: Secret channel
 - Channel not shown in `/list`, `/whois`, &c.
- **c**: Strip color
 - Remove color codes from channel messages
- **i**: Invite-only
 - Users must be explicitly invited to the channel (`/invite`)
- **m**: Moderated channel
 - Only users with `+v` may speak in channel.

More channel mode examples

[About](#)[Topology](#)[Interface](#)[Chatting](#)[Modes](#)[Services](#)[Netsplits](#)[SSL/TLS](#)[Questions](#)

- o <nick>: Channel operator
 - User is a *channel operator* (ChanOP; kicking, banning, &c.)
 - Name usually prefixed by the *hat* '@'
 - Often "off" on Freenode channels
- v <nick>: Voiced user
 - User can speak in moderated channel
 - Hat: +
- b <nick!ident@host> Banned user
 - User cannot join the channel

- IRC's default "registration"
 - Only checks for immediate usage collision
 - No authentication
 - Easily claimed by another
- IRC's default channel ownership
 - What happens when all ChanOps disconnect?
 - What happens when all users disconnect?
- *Services* fix these problems

- *Service* a program-controlled user, or *bot*, with special privileges
- Different nicknames for different purposes, examples:
 - NickServ: Account (nickname) management
 - ChanServ: Channel management
 - MemoServ: Offline user messaging
- Interact with them by direct messages
 - Example: `/msg NickServ help` for usage
- Like modes, different networks provide different services

- Create *accounts*, associate nicknames
- Register account: `register <password> <e-mail>`
- After connecting to the network, *identify* yourself with NickServ
 - Identify account: `identify <password>`
 - Auto-identify in irssi: `/NETWORK ADD -autosendcmd "/msg nickserv identify <pass>; wait 2000" freenode`
- Disconnect impersonators: `ghost <nick>`
- Auto-disconnect impersonators: `set enforce on`
- Remove enforcement hold: `release <nick>`
- Show account info: `info <nick>`
- Hide e-mail: `set hidemail on`
- Identify via certificate *fingerprint* (discussed later): `cert add`

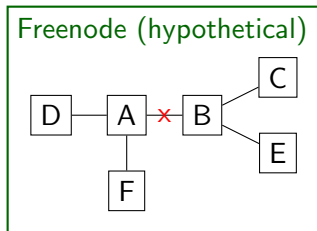
- Manage channels by *registering* them
 - Must be identified to services (i.e. NickServ)
 - `register <#channel>`
 - Register immediately after creation to prevent unmoderated channels
- Registered channels retain operator status
 - Set ChanOP mode: `OP <#channel>`
 - Unset ChanOP mode: `DEOP <#channel>`
- Set channel topic: `topic <#channel> <topic>`
- Get information about a channel with `/info <#channel>`

- IRC doesn't allow messaging offline users
- MemoServ can store & forward offline messages
- Cons: relatively unknown, some people don't notice memos immediately
- send `<nick> <message>` to send a memo
- read new to read new memos

Overview

[About](#)[Topology](#)[Interface](#)[Chatting](#)[Modes](#)[Services](#)[Netsplits](#)[SSL/TLS](#)[Questions](#)

- Servers in the network can become disconnected, resulting in a *netsplit*



- Example: servers A, D, and F separated from B, C, and E
- Users can only chat with their side of the netsplit
- Example in `irssi`:

```
13:16 -!- Netsplit *.net <-> *.split quits: finchd  
13:17 -!- Netsplit over, joins: finchd
```

- Wait out netsplits, or connect to different server

- Connections are *unencrypted* by default
- SSL/TLS connections are *encrypted* (the 'S' in HTTPS)
- TLS often offered on TCP port 6697
- Supply options to `/server` or `/connect`
- Server verification options:
 - `-tls_verify`: Verify server's certificate
 - `-tls_cafile`: Use specific Certificate Authority (CA); useful for self-signed certs
- Client verification options:
 - `-tls_cert`: Client's certificate
 - `-tls_pkey`: Client's private key

The End!

- Any questions?