

# Reflections on IRC, and Pain with Telegram

Brett Gilio

*<2020-09-18 Fri 21:11>*

## Contents

<b>1 Singing Praises, Crying Woes</b>	<b>1</b>
<b>2 Generational Differences</b>	<b>2</b>
<b>3 A note on IRCv3, and orcircd.</b>	<b>4</b>

## 1 Singing Praises, Crying Woes

IRC is one of those networking and communication protocols that ages well with the internet. Free software lovers, developers, and casual netizens (especially those born before the millenium) have created the infrastructure for discussing topics we are passionate about. Freenode and OFTC both have systematically helped structure not only project coordination and collaboration, but have created lasting ties between people as free software is an inherently social act. IRC represents freedom to many of us, and it is becoming unique again as other protocols such as XMPP slowly fall out of favor to the often proprietary, privacy invading, and untrustworthy competition.

This is where I disclaim that I use Telegram<sup>1</sup>. I was introduced to that platform a number of years ago, and was my first real connection point between like-minded developers and math nerds. Telegram is featureful, and able to sustain (and often surpass) parity with other platforms like Signal, WhatsApp, and other miseries. However, Telegram has been losing its esteem from my point-of-view over the last number of years.

Telegram suffers from too many problems: unnecessary features eerily resembling scope creep from the maintainers, over-engineering their network

---

<sup>1</sup><https://telegram.org/>

to support D.O.A. bloat such as TON (for reasons beyond my comprehension), and buying into the latest marketing buzz about blockchain'ing all-the-things. Additionally, the Telegram development has failed to keep its promises on many high demand features for years! For example, releasing their server-side source code.

My real big pain point with Telegram is that their funding model is cryptic, and relies on this unknown fortune of some “entrepreneur” named Pavel Durov<sup>2</sup>. Even the official website massages away all concerns about the source of funding by being generally blasé. They assure the users that they are not profiting off of user data<sup>3</sup>. How likely is this? I don't know. But the real question is, do you trust it? If Pavel Durov is arrested and all of his assets seized, what is the protection that Telegram will offer me from the Five Eyes (or whatever intelligence agency exists outside of that realm).

Additionally, Telegram does not guarantee any level of anonymity should you request it. To use Telegram you must register through a telephone number. If you are like Richard Stallman, you probably do not have a cellphone with MMS/SMS functionality. I do, but I do find it rather disconcerting that this is not optional to use the platform.

Between all of these apparent problems I have slowly but surely been moving to my other network of choice, IRC.

## 2 Generational Differences

I am rather young, but I am old enough to remember using AOL Instant Messenger and MSN Messenger on my Windows XP machine. Back then you were given particular constraints to work in. You had to understand your vulnerabilities of your machine (especially if you were using Windows), and you had to reason with the constraints offered by the protocol you were messaging on.

In those days, I was introduced to programming via Visual Basic 6 and C++. My go to hobby was writing exploits for AOL's OSCAR<sup>4</sup> and TOC<sup>5</sup>. These exploits took advantage of specific vulnerabilities in message handling, DCC exchanges, and lack of flood-prevention to cause some nuisance effect to others on those platforms. The fun of this, besides being a nuisance to others, was that you were forced to reason with the limitations of those

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Pavel\\_Durov](https://en.wikipedia.org/wiki/Pavel_Durov)

<sup>3</sup><https://telegram.org/faq#q-how-are-you-going-to-make-money-out-of-this>

<sup>4</sup>[https://en.wikipedia.org/wiki/OSCAR\\_protocol](https://en.wikipedia.org/wiki/OSCAR_protocol)

<sup>5</sup>[https://en.wikipedia.org/wiki/TOC\\_protocol](https://en.wikipedia.org/wiki/TOC_protocol)

networks.

IRC is similar (except most IRCd implementations are FAR more secure and not as easy to exploit with a good client) in that you are set with some limitations. IRC is generally not going to handle your images, distribute some minified JS to give you some animated sticker pack, etc. What IRC excels at is cheap, reliable infrastructure. This is different from your Telegram, WhatsApp, Signal, or (RMS forbid) Discord. While Slack and Matrix are in varying degrees based on the IRC protocol even their extensions have severely limited the variations of clients that get first-class support.

This is in great contrast to IRC where well crafted clients are available in just about every crevice of your (mostly) standard GNU/BSD/Plan9 system. Even Emacs comes pre-packed with two (maybe three, who knows) IRC clients<sup>6,7</sup> built into it with many more that can be fetched and plugged<sup>8</sup>.

Drew DeVault wrote an article<sup>9</sup> which exemplified wonderfully that the “lack of features” on IRC are a positive, alluding to how poorly and/or over-engineered your other choices are today.

For most of these features, I think that people who have and think they need them are in fact unhappier for having them. What are some of the most common complaints from Slack users et al? “It’s distracting.” “It’s hard to keep up with what people said while I was away.” “Threads get too long and hard to understand.” Does any of this sound familiar? Most of these problems are caused by or exacerbated by features which are missing from IRC. It’s distracting because your colleagues are posting gifs all day. It’s hard to keep up with because the infinite backlog encourages a culture of catching up rather than setting the expectation that conversations are ephemeral.

– Drew DeVault

This is further complimented with the fact that it takes leaps-and-bounds more effort to port your Discord client to some unconventional (possibly custom) operating system. To port an IRC client, it is much more trivial and at the least you can get to communicate more quickly because the protocol is just not nearly as convoluted or cumbersome to wrestle with.

---

<sup>6</sup>[https://www.gnu.org/software/emacs/manual/html\\_mono/erc.html](https://www.gnu.org/software/emacs/manual/html_mono/erc.html)

<sup>7</sup>[https://www.gnu.org/software/emacs/manual/html\\_mono/rcirc.html](https://www.gnu.org/software/emacs/manual/html_mono/rcirc.html)

<sup>8</sup><https://github.com/jorgenschaefler/circe>

<sup>9</sup><https://drewdevault.com/2019/07/01/Absence-of-features-in-IRC.html>

### 3 A note on IRCv3, and orcircd.

This short article wouldn't be complete without making a few tangential remarks. First, IRCv3 is a work in progress to assist in creating a modernized next-generation IRC protocol standard. Some parts of this are ratified, many are still in draft phase. Some of the drafts are absolute lunacy, and others are quite nice (I do not wish to specify here, lest I make enemies).

With the IRCv3 protocol, and my recent turn to IRC in general I have been doing research on various IRCd implementations out there as well as reading the RFCs. IRC is absolutely fascinating! I have started somewhat of a research pet project to combine the latest innovations of OCaml and take standard IRC and IRCv3 specifications to create my own IRCd. With this I have had a lot of help from friends who are interested in contributing, as well as the OCaml community, and many Freenode staffers. Hopefully something comes of this.

Here is a list of various IRCd server implementations:

- inspired (C++)
- oragono (Go)
- unrealIRCd (C)
- ircd-seven (C)
- miniircd (Python)
- matrix-ircd (Rust)
- orcircd (OCaml) *Work in progress*

---

### Have a response?

Responses and discussion pertaining to any of the blog entries on my website are welcome! Start a discussion on the mailing list by sending an email to [~brettgilio/blog-discussion@lists.sr.ht](mailto:~brettgilio/blog-discussion@lists.sr.ht).

### Errata:

- *<2020-09-23 Wed 11:31>* Fix dead links and grammar.