Mosh: the Mobile Shell

Tom Ryder
tom@sanctum.geek.nz
https://sanctum.geek.nz/

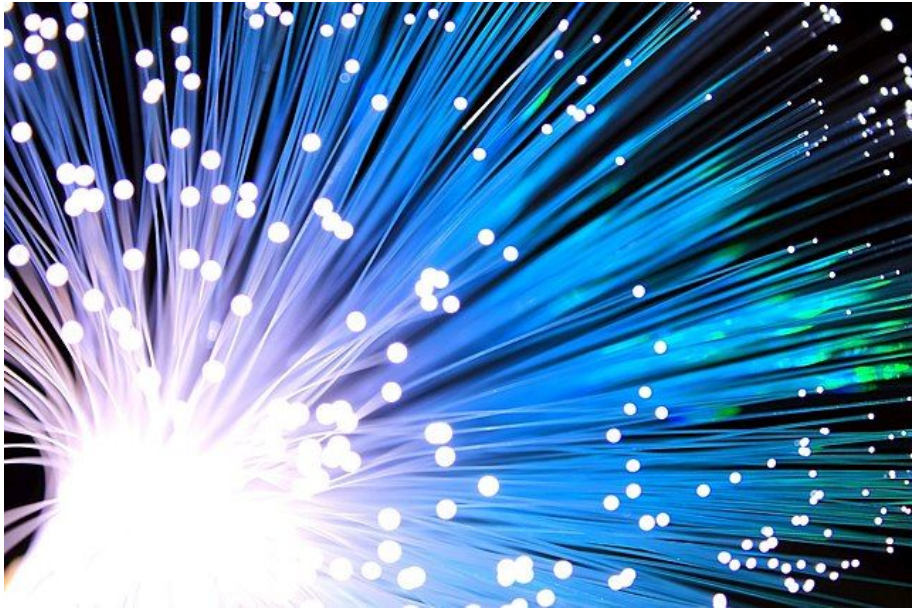
FOSS pilgrimage—1/2

- In **February 2023**, I travelled to **London, England**…
  - …and then to **Brussels, Belgium**…
  - …for the **FOSDEM conference**!

FOSS pilgrimage—2/2

- **8,000+ fellow hackers**, **700+ great talks**…
- Nerds—err…I mean…*free and open source software advocates*—everywhere!
- Had a great time! Going back next year.
- There was *just one problem*…

Constant nagging—1/2



- I was **18,000km** away.
- Latency home *sucked*.
- Even fiber optics can only go so fast.
- There’s not much one can do about the **speed of light**…

Constant nagging—2/2

- I was trying to use IRC, check travel information, and read mail **over SSH**.
- The SSH server was my NUC on my desk, at home in NZ.
  - Type a character…
  - …wait a second…
  - …it should show up…
  - …err, probably…

Latent white cloud—1/3

- I live in New Zealand. The land of the long white cloud…
- …and the **weirdest time zone** (UTC+12/13).
- Incomprehensible to Americans…
  - Seriously, try scheduling streams with a few Americans sometime.

Latent white cloud—2/3

- From NZ, most of the big cheap computers are **far away**—at least as far as Australia.
- So: I’m used to **high latency** for **batched** jobs.
- It doesn’t really matter if it takes 400ms for a command to reach my VPS.
- I didn’t *type it* on the VPS, after all.

Latent white cloud—3/3

- But *typing* commands and messages on a remote server—over a slow international SSH link—is way more painful.
- You don’t realise how important that **immediate feedback** is until you don’t have it.
- You press the key. The character appears.
- We don’t know how lucky we are…

In the gloamin’—1/2

- To make matters worse, I was **roaming**—moving **between networks**—a *lot*.
  - In-flight WiFi…
  - Roaming 3–5G in multiple countries…
  - Hosts’ home WiFi networks…
  - Café hotspot internet…
  - Hotel hotspot internet…
  - Workplace VPN, WireGuard, Tor…
  - Hibernating the laptop to save power…

In the gloamin’—2/2

- I kept **disconnecting**, and having to reconnect.
  - That’s the nature of the TCP beast.
  - You can’t just **resume** the same connection from another network.
  - Gotta suffer that original SYN.
- **tmux** definitely helps here…
  - …but it’s still annoying having to reconnect.

The mother of adoption—1/2

- I remembered reading about **Mosh**, which has been around since 2012.

  (Yeah…nobody's ever called me an "early adopter"…)

- Mosh describes itself as a **mobile shell**.
- It purported to solve all of my problems.
- It's **free software**.

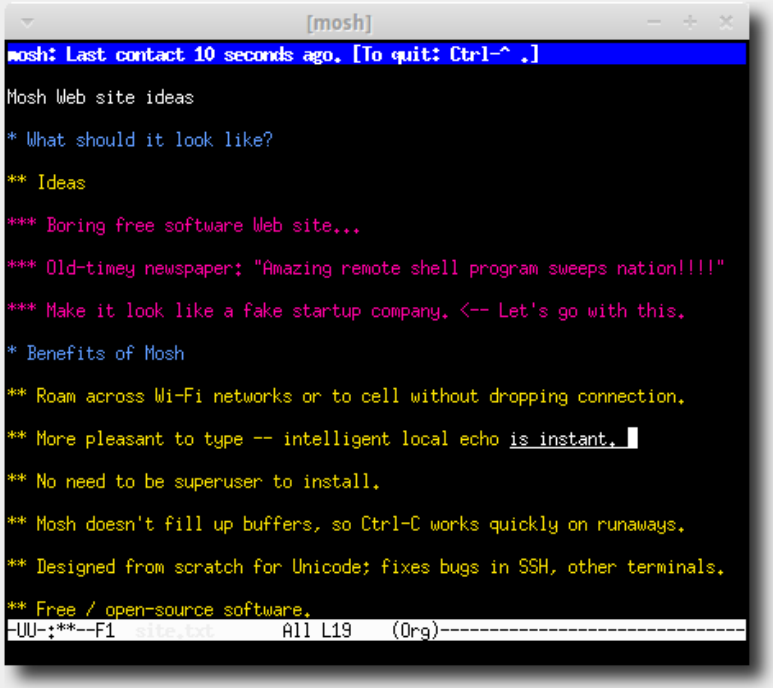
Mosh

(mobile shell)

Remote terminal application that allows **roaming**, supports **intermittent connectivity**, and provides intelligent **local echo** and line editing of user keystrokes.

Mosh is a replacement for interactive SSH terminals. It's more robust and responsive, especially over Wi-Fi, cellular, and long-distance links.

Mosh is free software, available for GNU/Linux, BSD, macOS, Solaris, Android, Chrome, and iOS.



Getting Mosh » Tech Video »

The mother of adoption—2/2

- So, I installed it!
  - **Debian GNU/Linux:** `$ sudo apt install mosh`
  - **Termux (Android):** `$ pkg install mosh`
- **Forward/whitelist ports 60000–61000** on the remote end.
  - You may not need to do this, depending on if you have a firewall.
- And then:

  `$ mosh remote`
- **Done.** After the port forwarding, it Just Worked™.

Demo—Simulated bad network

- We'll simulate a bad network connection with *tc(8)*, using a `netem` **queuing discipline**.

```
$ sudo tc qdisc add dev wlan0 root netem \
    delay 1000ms 500ms distribution normal \
    loss 10% 50% \
    corrupt 2%
```

  - Between 0.5–1.5 seconds of packet **delay**.
  - Packet **loss** of around 10%, with series of lost packets likely.
  - 2% packet **corruption**, just to really mess things up.

Demo—SSH—1/2

- Keystrokes will appear on screen with screenkey, to demonstrate latency.
- We'll start with a connection via **plain SSH**.
- All the keystrokes show in the end…
- …but it's *slooooow.*
- Hard to type even short passages.

Demo—SSH—2/2

- Now let’s switch networks, from local WiFi to my mobile hotspot, to simulate **roaming**.
- SSH uses a TCP stream with a defined pair of connection endpoints.
- It can’t cope with this.
- We have to **reconnect**.

Demo—Mosh—1/2

- Type a bit, and keystrokes show *immediately*!
- No, Mosh isn’t bending space and time.
- The keystrokes are still taking the same amount of time to cross the network.
- Mosh *guesses* that text will appear there shortly:
  predictive local echo.
- Predicted text is underlined before it’s confirmed.

Demo—Mosh—2/2

- Let’s switch back to the venue WiFi now.
- Mosh might show its blue bar for a bit, maybe…
- …but it just starts throwing UDP packets at the endpoint…
- …and soon we’re connected again.

Reconnect in the summer

- You can sleep or hibernate a laptop with Mosh running.
- It will reconnect when you start the computer back up again, losing no state.
- If you’re used to using SSH, the first time you see this, it’s like magic.

Deep magick—1/4

- Mosh does its **authentication** with SSH, but then switches to its own protocol for the rest of the session.
  - Your ~/.ssh/config file will keep working.
  - Still using modern high-strength encryption.
- Uses UDP instead of TCP.
  - “Connectionless”—just picks a UDP destination port on the remote end, and remembers it for that run.
  - Sends packets to that port from wherever it is presently.

Deep magick—2/4

- A terminal emulator's *complete screen state* runs on the remote server…
- …not just a stream of characters, as in SSH.
- This means Mosh can set a **remote terminal frame rate**.
- So Ctrl+C always works, even if you *cat(1)* a huge file!

Deep magick—3/4

- Mosh uses a custom protocol—**State Synchronization Protocol**—designed for this purpose.
- It remembers the last host it heard from with the chosen private key.
- It sends the changes to the *screen* back over the wire… not just a sequence of characters for the local terminal to interpret.

Deep magick—4/4

- Watches screen state changes in response to keystrokes.
- If the same keystrokes as you’re typing start appearing in a line, it guesses you’re typing, and turns on its local echo.
- Usually only needs a few characters, and your keystrokes show up instantly.

Interactive only

- Mosh’s benefits are designed for **interactive** terminal use.
- There’s no point running other programs with it in **batch** mode.
- It doesn’t implement SSH’s **tunnelling** or **forwardings**.
- Just use SSH in both cases.

Downsides?—1/3

- You have to make UDP ports **60000–61000** whitelisted (and **forwarded**, if behind NAT) on the remote server.
  - Might be a hassle in some limited contexts.
  - You can request specific ports or ranges if needed.

Downsides?—2/3

- Mosh leaves dead server processes lying around if the client vanishes, with no default timeout; they run until explicitly terminated.
  - Fix this by setting `MOSH_SERVER_NETWORK_TMOUT` on the remote end, in `~/.pam_environment`.
  - I do **24 hours (86400)**.
  - Anything I need alive longer than that should live in a **tmux session**, anyway.

Downsides?—3/3

…err, **that’s about it**.

- Free software
- Behaves like interactive SSH
- Uses existing SSH setup (password, keys, config…)
- Packaged for everything (even Windows, sort of)
- Doesn’t require root
- Spotless security record

Technical detail

- USENIX presentation (youtube.com, 11m42s)
  - Presentation overview of how Mosh works.
- The Mosh research paper (PDF, 172 KiB)
  - Lots of technical detail about the advantages and how it’s all implemented.
- Mosh: A State-of-the-Art Good Old-Fashioned Mobile Shell (PDF, 452 KiB)
  - An essay with even more detailed information than the above, including some design philosophy.

Skanking

- This isn’t Mosh’s logo, by the way.

  (It doesn’t seem to have one.)

- I just thought the **moshing** (“skanking”) man was funny.
- It’s actually 80s crossover thrash band Dirty Rotten Imbeciles’ logo.

Questions?

Mosh website

Email: tom@sanctum.geek.nz
Website: https://sanctum.geek.nz/
Fediverse: @tejr@mastodon.sdf.org