Dial-up revisited: Why it’s needed and how to run an oldschool ISP

Özcan Oğuz

FOSDEM 2026 - Brussels

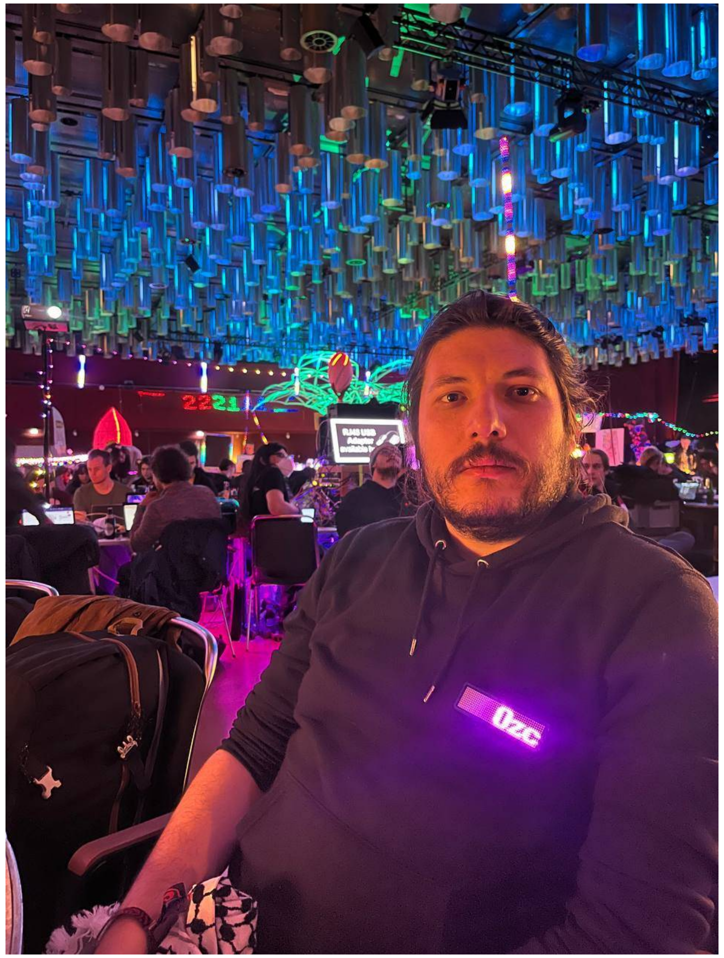
1st February 2026

BY SA

Özgür
Yazılım
Derneği

Kullan, araştır, paylaş, geliştir!

Who am I?



Özcan Oğuz
Özgür Yazılım Derneği
(Free Software Association in Turkey)
ozcan@oyd.org.tr

- Free software (h)ac(k)tivism
- Solution + software developer
- A tinkering & hacking enthusiast

This talk aims to cover:

- The basics of dial-up connections
- Why do you need it in 2026
- How to set up a personal dial-up ISP using free software
- Known limitations

What was dial-up?

- It was *de facto* the main connection method to the Internet back in 90s and early 2000s until the (A)DSL era
- Uses the old plain conventional telephone network (PSTN) as infrastructure
- A modem is required to turn data signals into telephone signals or voice!

The advantages of dial-up

- It requires **no additional infrastructure** than the telephone line
- In some cases, even a subscription is not required! (e.g 146 in Turkey)
- This also (mostly) means that providing Internet access have the same prerequisites

Issues

- Dial-up connections had two major issues: Low bandwidth and respectively high cost.
- Theoretical highest bandwidth is only 56 kbps download and 40 kbps upload (V.90-V.92 @ 2000)
- You have to pay an amount for the Internet access to the ISP, and also for the **telephone call**
- Usually users were not online all the time: Connect, get the response and hang up

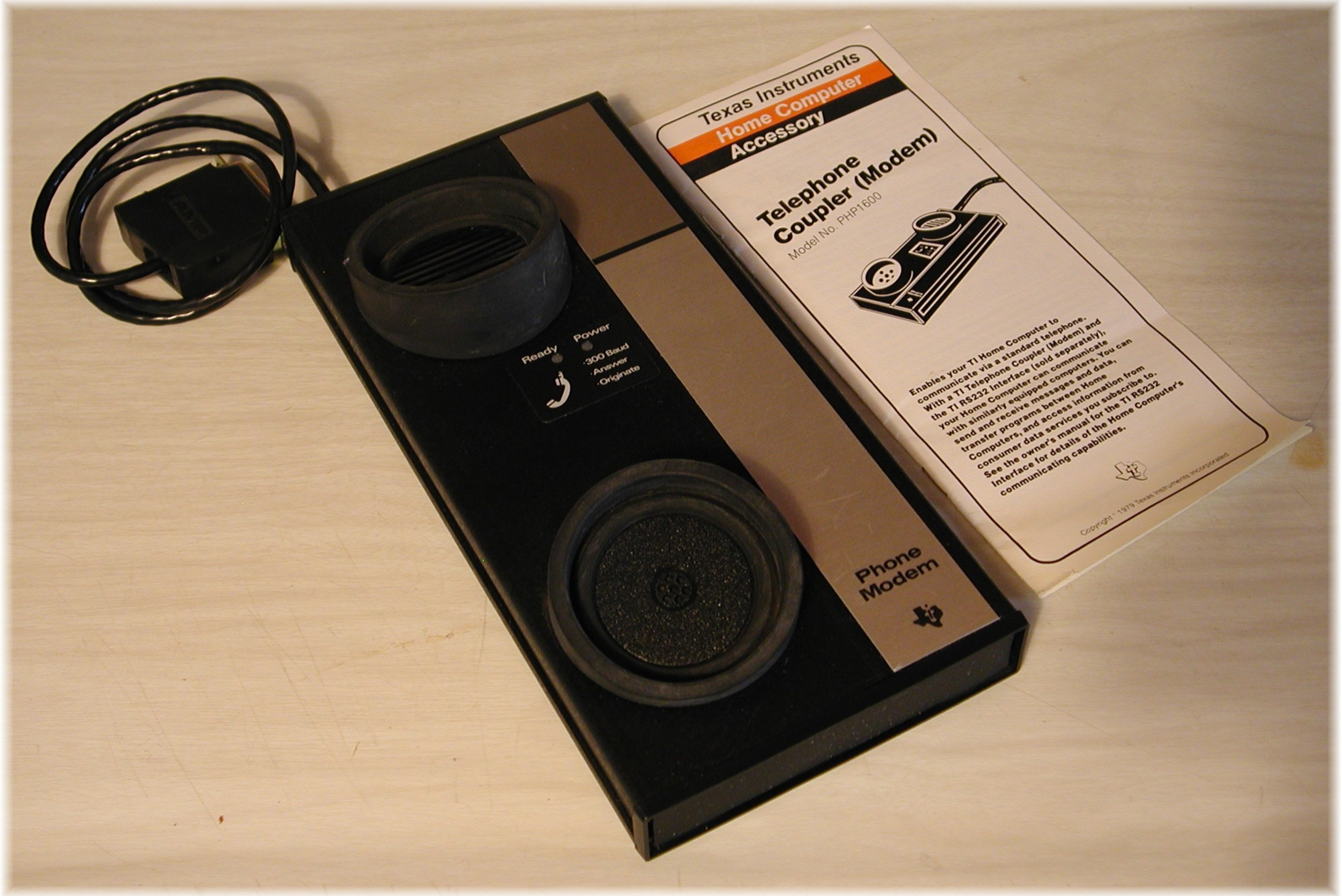
Decline

- Starting from 2000s, dial-up is replaced with **broadband** connection methods like DSL, cable, FTTx and of course **mobile broadband**
- Most of the countries have shut down their dial-up services in 2010s
- Notably, AOL shut it down **only four months ago**, while more than **150.000** people are still using it
- In most places, even PSTN is in decline

So, why do I need dial-up in the year 2026?

For two reasons:

1. It is sometimes the only way to connect an old/retro device to the Internet
2. Dial-up does not require a complex infrastructure, if you have phone you will have connection.



Texas Instruments
Home Computer
Accessory

Telephone Coupler (Modem)
Model No. PHP1600

Enables your TI Home Computer to communicate via a standard telephone. With a TI Telephone Coupler (Modem) and the TI RS232 Interface (sold separately), your Home Computer can communicate with similarly equipped computers. You can send and receive messages and data, transfer programs between Home Computers, and access information from consumer data services you subscribe to. See the owner's manual for the TI RS232 Interface for details of the Home Computer's communicating capabilities.

Copyright © 1979 Texas Instruments Incorporated

Dial-up can be savior

Thus, it can be used as a tool to **circumvent censorship**.

During the 2011 Egypt uprising, the Internet was under complete blackout.
But **telephones** were still up. People were using dial-up and radio comms.

Why do I started this research?

Turkey has similar throttles all the time.
Payphones still exist everywhere and so many places have landlines.

I started a project called **neo146**, it is an information gateway so far, but I want it to be usable as a dial-up ISP.

The documents and guides I’ve found online was either outdated or requires hardware that I cannot find.

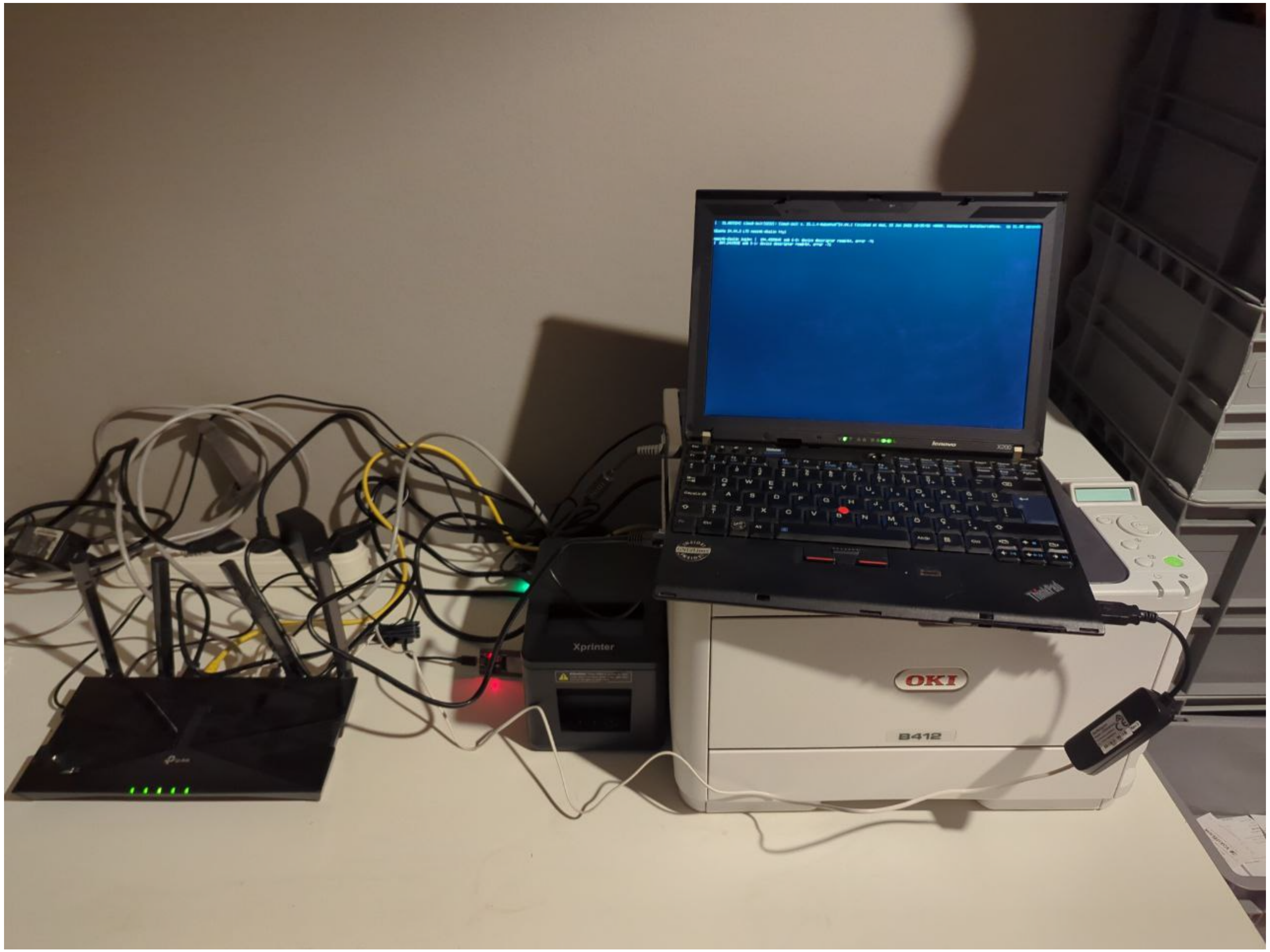
So, to begin with...

You will need:

1. A modem for the ISP dial-in (it must be a hardware modem!)
2. A computer to be the dial-in server (runs a GNU/Linux distro)
3. A modem for the client (hardware modem is strictly required)
4. Landline, analog telephone adapter, PBX or similar to provide telephony on both ends

My hardware stack on the server (in Istanbul)

- ThinkPad X200 with Libreboot
- SIP subscription from a Turkish VoIP company
- TP-Link VDSL modem telephony feature
- StarTech.com USB56KEMH2 hardware modem *(not sponsored!)*


lenovo
X200
ThinkPad
Xprinter
OKI
B412

My hardware stack on the client

- This very laptop computer, ThinkPad X270
- SIP subscription from the same Turkish company
- Cisco Linksys SPA-3000 ATA (since we do not have a landline here in ULB)
- The same modem

Software to be used

- Ubuntu Server 24.04 LTS on server, Xubuntu 24.04 LTS on client
- `ppp` on both client and server to establish the connection
- `mgetty` for modem and call management on dial-in server
- `wvdial` for dial-up connection on client

Server setup

1. Install ppp and mgetty. `sudo apt install ppp mgetty`
2. Connect your modem to the computer and find the device path. (e.g `/dev/ttyACM0`)
3. Create a systemd service for mgetty

(`/etc/systemd/system/mgetty.service`)

```
[Unit]
Description=dialin-server
Requires=systemd-udev-settle.service
After=systemd-udev-settle.service

[Service]
Type=simple
ExecStart=/sbin/mgetty /dev/ttyACM0
Restart=always
PIDFile=/var/run/mgetty.pid.ttyACM0

[Install]
WantedBy=multi-user.target
```

Server configuration (mgetty)

We need to edit `/etc/mgetty/mgetty.config` file.
mgetty configuration can vary depending on what modem are you using. Look at the man page and docs, also your modem datasheet.
Also we need to comment the line starting with an asterisk (*) in `/etc/mgetty/login.config` to disable fax fallback.

```
debug 7

port ttyACM0
  speed 115200
  data-only yes
  toggle-dtr yes
  toggle-dtr-waittime 500
  ignore-carrier no
```

Server configuration (ppp)

To configure ppp, we need to change the file `/etc/ppp/options`:

```
ms-dns 1.1.1.1
asyncmap 0
auth
crtscts
lock
show-password
passive
silent
+pap
debug
proxyarp
lcp-echo-interval 0
lcp-echo-failure 0
noipx
novj
noccp
nopcomp
noaccomp
```

Server configuration (ppp)

For device options in ppp, we need to create a file at
`/etc/ppp/options.{your_device_path}`

```
local
lock
nocrtscts
192.168.32.1:192.168.32.105
netmask 255.255.255.0
noauth
proxyarp
lcp-echo-failure 60
```

Server configuration

- Create an user for the authentication, in dialout group with the shell `/usr/sbin/pppd` and set a password for it
- Append this user credentials in `/etc/ppp/pap-secrets` file with the format `username * "password" *`
- Enable IP forwarding `net.ipv4.ip_forward=1`
- Enable masquerading and allow ufw to forward traffic

The notes on VoIP

VoIP is **hostile** to the modems.

Modern VoIP codecs and supportive technologies are designed for voice transmission, thus they will break the data communication.

In ATA or any VoIP backed setup, we need to cancel all kind of ”fancy” stuff, such as echo cancellation, adaptive jitter buffer, VAD, T.38, and any other codec than G.711.

The notes on VoIP (continues)

56 kbps was a dream, we do not have this infrastructure anymore...

Even if you use PSTN, sometimes (or mostly) the providers switches your call via VoIP or apply compression, and this makes running in those speeds impossible.

The theorical limit is 21100 baud, in reality 14.4k on VoIP is **GREAT**.
(Also your V.90 modem has 33.6k uplink, so 56k was never an option...)

Client setup

1. Install ppp and wvdial `sudo apt install ppp wvdial`
2. Add your user to `dialout` group
3. Connect your modem and learn it's path (mine is `/dev/ttyACM0` again)
4. Run `wvdialconf`, if it detects your modem it's fine
5. Edit the autogenerated file `/etc/wvdial.conf` for fine tuning

Client configuration

```
[Dialer Defaults]
Modem = /dev/ttyACM0
Baud = 115200
Phone = 08502420146
Username = neo146
Password = neo146
Stupid Mode = yes
Carrier Check = no
New PPPD = yes

Init1 = ATZ
Init2 = AT&F
Init3 = AT+FCLASS=0
Init4 = AT+MS=V22B,1,300,2400
Init5 = ATM1L3
```

Client configuration

Now, edit the ppp configuration to make our computer be able to access to the Internet via ppp0:

```
115200
lock
crtscts
local
noipx
novj
noccp
nopcomp
noaccomp
lcp-echo-interval 0
lcp-echo-failure 0
defaultroute
replacedefaultroute
usepeerdns
```

And now, the moment of glory

- Before connecting the dial-up, do not forget to close the applications that poll a server constantly, for notifications or updates, such as Thunderbird or IM clients
- Open a terminal window on your client and type `sudo wvdial`
- It will dial your ISP server and connect you to the Internet!

Demo time!

OK I have connected, now what?

- Dillo for graphical, lynx or links for text-based browser
- Wiby.me for search engine
- You can build your own plain HTML portals/services
- **Do not use HTTPS** because it will probably not work
- Set up your own **proxy**, using tinyproxy or squid (or something else)

Notes on improving the connection stability

- On V.32, we can use MTU/MRU as 512, on V.22b drop to 296
- Update client IPv4 configuration to improve your connection

```
sudo sysctl -w net.ipv4.tcp_sack=0
sudo sysctl -w net.ipv4.tcp_window_scaling=0
sudo sysctl -w net.ipv4.tcp_timestamps=0
```

What is next on my research?

- I am currently running a research on implementing native dial-up support on Android within the **neo146** project, since most phones have telephony internally
- Acoustic coupler demos and tweaks to make it usable with payphones
- More stable connection on server side
- A modern version of GNOME PPP, it is not updated since 2006
- I had no chance to update it more, but I want to provide a low bandwidth infogate to users via neo146
- I want to provide dial-up and MUD services if I can cover the costs

Further reading and thanks

- My English website *under construction* → `https://ooguz.dev`
- Neo146 project website → **`https://neo146.net`**
- GitHub → **`https://github.com/ooguz/neo146`**

- Special thanks to Doge Microsystems and dialup.world for a starting point for this project

Thank you for listening!

Any questions?

Contact me

Özcan Oğuz
Özgür Yazılım Derneği
ozcan@oyd.org.tr

Twitter (and most places): **@ooguz**
Fediverse: **@ooguz@chaos.social**
Web: **ooguz.dev** / **www.ozcanoguz.com.tr**
GnuPG: **0x3d975818**
XMPP+OMEMO: **oo@5222.de**