

Network: Home-Lab project [OLD].

Comment [READ IT FIRST]

This entry has become relatively outdated. After more than a few years of a configuration like the one below some things started to, annoy or simply require too much maintenance. Currently the core of the network itself looks the same, I still use cloudflare for some of the websites I host (only the less important ones now), I still use tailscale although I also try to have alternative network access in case tailscale decides to remove Free Tier.

The things that have changed the most are hosting `wiki.ostrowski.net.pl`, `ostrowski.net.pl` and a few other related services like mail and calendar. Every time the internet switched over to the backup ... it was the cloudflare tunnel that displayed for several minutes that the page was not available, despite the fact that it should have it in cache. Power outages have also been a problem, where I live there have been no problems with the power supply for years until one day a housing estate started to be extended, since then the quality of the power supply is not at the level it was before. All this has meant that these sites are now hosted on OVH's VPS, less of a problem less of a pain, admittedly I still back them up to my home.

Hosting mail on an IP address that is in the residential range is almost impossible You always get caught up in spam lists, which you can't get out of because the verification email doesn't even arrive. So I used OVH MX5 hosting, i.e. 5 email accounts of 5GB each connected to my domain on OVH, it works great DKIM and SPF configured and I have never fallen into another spam list since.

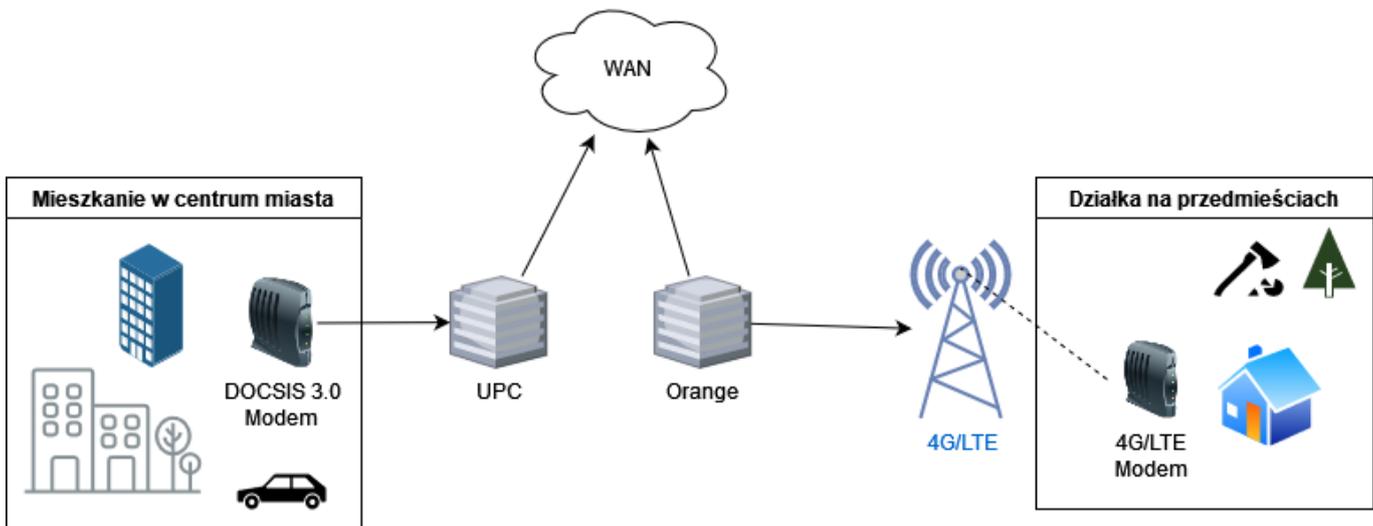
It's no longer full self-hosting, which I'm not too proud of, instead the setup has changed to a mixed with two sites, only that one of them is a VPS 😊

Requirements and specifications

My home network is a constantly changing structure depending on what I happen to be doing for a particular client or what I happen to be interested in, and I decided to test this on a living organism. As for my requirements, they are not great, the network will have at most a few users, one of which is

me 😊 . The network needs to be very configurable, with some services made available via tunnels and others via port forwarding and reverse proxy. Some services must be accessible from the outside, to me but also to clients and to the people I am training. Another aspect that is important to me is cost optimisation, equipment must be as cheap as possible so that it can be easily replaced. A lot of the components of the network I am going to present to you here are home-grade equipment or have been bought by me (for a pittance) from my former clients.

Network location diagram



Requirements

Requirements for my network:

- public IPv4 address (can be dynamic).
- no blocking of tunnels or IPsec from ISPs.
- easy and quick configurability
- possibility of setting up a VPN tunnel between two locations (for monitoring only)
- hosting services on different domains
- equipment and systems with good documentation
- low costs

Specification

ISP

As for my ISP, it is PLAY (formerly UPC). Every UPC subscriber gets a dynamic IPv4 address, with a lease time of about a month, so this exhausts my IP address requirements. The modem provided by UPC is a little box called „Horizon”, it's a regular DOCSIS 3.0 standard modem ([Bardzo nice resource that shows how DOCSIS 1.0/1.1 works](#)).

Routers, Switches and other network devices

<p>Picture:</p>				
<p>Description</p>	<p>MikroTik hAP AX lite</p>	<p>TPlink TL-SG105E</p>	<p>Mikrotik mAP lite</p>	<p>Mikrotik RB941</p>

Name	LAB-ROUTER	TP-LINK	LAB-AP	MikrotikDzialka
-------------	------------	---------	--------	-----------------

As I mentioned in the introduction, these are not enterprise-class devices, but for my needs they are sufficient and allow a high degree of configurability.

Network diagram

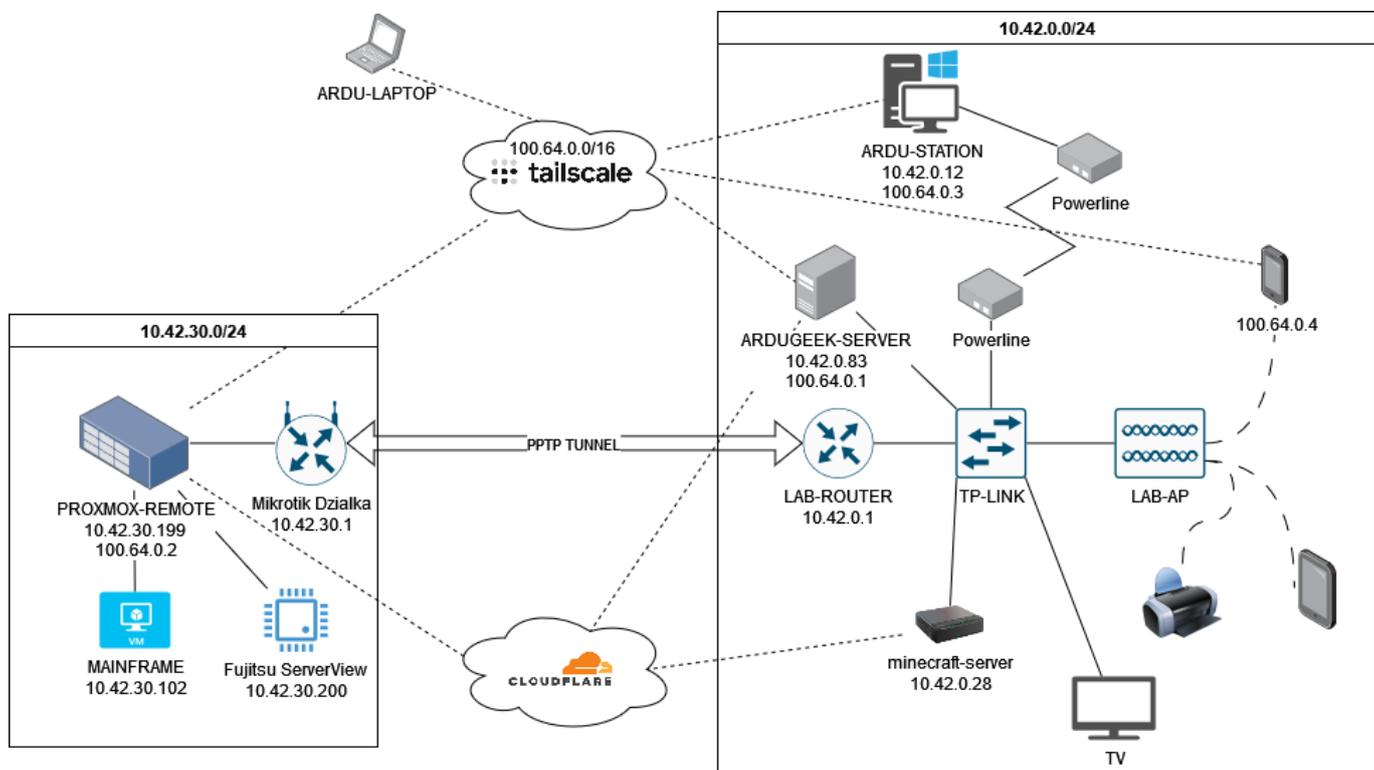


Diagram description

As you can see above, the network is not very complex, but it has several main components such as servers and routers in both locations. In the following sections, we will discuss the individual components of the overall scheme.

LAB-ROUTER

This is the main router where all port forwarding to the relevant servers takes place. It used to be a VPN server as well, now it just gets a PPTP tunnel to monitor the devices in the other location. The port forwarding is to services on minecraft-server and ardugeek-server. I am using the DDNS feature available from mikrotik on it as not all services want to be on cloudflare tunnels.

Mikrotik Dzialka

This is just a simple client for 4G/LTE networks as a DHCP client with a gateway. It is configured as a PPTP tunnel client. I'll mow this tunnel down for today only to monitor the devices. Everything else is done by tailscale. Below is the equipment I use for 4G/LTE connectivity



TP-LINK TL-MR100
Source: [Morele.Net](#)



4G/LTE antenna
Source: [Morele.Net](#)

PROXMOX-REMOTE - Physical server



The physical server is an old Fujitsu Primergy RX300 S8 server that I bought back from one of my clients a long time ago. I have hypervisor installed on it [Proxmox VE](#). It is not a server with an exorbitant specification but it is sufficient for various experiments. It has an Intel Xeon processor, 32GB of ram, Two arrays, one RAID1 (256GB SSD) and the other RAID5 (3.5TB HDD). A tailscale

service is running on it.

It is also the exit node for the 10.42.30.0/24 network, so if any of the Tailscale network clients want to connect to anything on that network it will exit to it through this server.

MAINFRAME - Virtual machine on proxmox-remote



Logo of the Xubuntu distribution

Source: [wikimedia.org](https://commons.wikimedia.org/wiki/File:Xubuntu_22.04.1_Linux_Live_Environment_Logo.png)

This is a virtual machine running [xubuntu](#) 22.04. On it I have services running such as:

- <https://ai.ardugeek.ovh/> - Open Web UI
- <https://jupyter.ardugeek.ovh/> - Jupyter lab
- <https://kasm.ardugeek.ovh/> - Kasm workspaces
- <https://portainer.ardugeek.ovh/> - Portainer
- <https://files.ardugeek.ovh/> - File browser
- <https://shell.ardugeek.ovh/> - Shell in a box

All of these services are hosted using a cloudflare tunnel. I don't have a public address available at this location on my sim card, so unfortunately only the cloudflare tunnel remains. These are not

critical services for me so if the tunnel disconnects (and it tends to once in a while 😊) it doesn't affect the performance of my network. Connector tailscale doesn't run on this VM it just runs on the hypervisor itself.

ARDUGEEK-SERVER - Physical server



This is a physical computer that I assembled some time ago from ordered components, it has a slightly outdated specification by today's standards but for my needs it is 100% sufficient. The system

installed on it is Windows Server 2019. The specifications of this server are: Intel i3 3rd generation, 16GB of ram, and many different types of external and internal drives connected to it. Many of these are backed up to different locations (other drives or onedrive) using a tool [Duplicati](#) with retention and differential. It is the focal point of my entire network, it is the server where it all started (not this particular one of course, there have been many predecessors with the same name), it is where I connect remotely when I am somewhere out of the house and something breaks and I need to fix something remotely, it is where I test various solutions, it is where this Wiki is running. Everything in my network is somehow connected to this server.

It is also the exit node for the 10.42.0.0/24 network, or all tailscale traffic. It allows clients of the Tailscale network to go out into the world through this computer.

Below is a summary of the services running on this server.

Internal services not available to the public:

- [Node-Red](#)
- [Microsoft SQL server](#)
- [Adminer](#)
- [Duplicati](#)
- [WakeMeOnLan](#)

Services available to the public:

- <https://start.ardugeek.ovh/> - Homepage for my services, static HTML/CSS.
- <https://ostrowski.net.pl/> - My business card page with training information and contact information for me, static HTML/CSS
- <https://wiki.ostrowski.net.pl/> - The wiki you are reading now configured with the package [DokuWiki](#)
- <https://filevista.ardugeek.ovh/> - File server used for my private purposes but also for clients, configured with the tool [FileVista](#)
- <https://plex.ardugeek.ovh/> - server [plex](#) for multimedia. (I'm probably one of the last customers to buy media from an empik on disc or in MP3/MP4 format 😊)
- <https://audiobook.ardugeek.ovh> - [Audiobookshelf](#) application to listen to audio books
- <https://auth.ardugeek.ovh> - [A script](#) of my own making it possible to connect securely via NAT translation to services.

ARDU-STATION - physical computer

This is my main workstation and the computer on which I happen to play video games. The specs are Ryzen 5 5800U, 32GB ram and 1TB SSD, 1TB HDD, 2TB HDD. This is the main workstation with all the environment for virtualisation and testing new solutions this is where I couch the Hyper-V and CHR environment from MikroTik for training. It is here that the [Hyper-V topology mapping script](#). This is also where I have the tailscale connector installed because this is also where I connect to my servers and test various solutions, and in my spare time I like to play genre classics.

There are no services as such here, I can only mention one in total, namely I have WSL2 configured here so that I can run a program on it [GNU-RADIO](#) for simulating signal processing.

minecraft-server - physical server



This is a very old industrial mini pc from giada. It only has 8GB of ram and an intel i3 3gen. Here I have the package installed [Application Management Panel from CubeCoders](#), I host servers here for gaming with friends and family. Mainly running a multi-platform Minecraft server here at the moment. I have a Cron job running here that every two days packs the entire Minecraft server into a ZIP and uploads it to ardugeek-server. You can see the configuration of my server [here](#).

The only thing extra here besides the standard Minecraft server configuration is that the cloudflare tunnel provides the following sites:

- <https://amp.ardugeek.ovh> - Minecraft server management panel
- <https://bluemap.ardugeek.ovh> - plugin [BlueMap](#) for displaying the server map in the browser

Bluemap plugin

BlueMap is a program that reads your Minecraft world files and generates not only a map, but also 3D-models of the whole surface. With the web-app you then can look at those in your browser and basically view the world as if you were ingame! Or just look at it from far away to get an overview.

Z <https://bluemap.bluecolored.de/>

This is a recent discovery of mine and I think this plugin should be available to everyone who has their own Minecraft server. This plugin allows you to view the game world in 3D as if you were in the game. Luckily, the plugin is written in JavaScript so it doesn't load the server. It looks spectacular. Installation is simple, and the port on which the map is visible is 8100.

Tailscale principle

Tailscale is a VPN service based on a mesh architecture that simplifies the connection of multiple devices in a secure private network (tailnet). Instead of the traditional hub-and-spoke model (where all traffic flows through a central server), Tailscale enables direct, encrypted point-to-point connections between network nodes. Each connection is end-to-end encrypted using the WireGuard protocol (with the ChaCha20-Poly1305 algorithm), and the private keys remain exclusively on the users' devices. The network nodes exchange keys and metadata through a control plane - the Tailscale coordination servers - using the Noise IK protocol (X25519) wrapped in TLS, which enables authentication and configuration distribution (ACL).

With NAT traversal mechanisms, each Tailscale client attempts to break through the firewall (hole-punching) using a customised UDP protocol. If a direct connection cannot be established (e.g. due to symmetric NAT), Tailscale automatically uses DERP (Designated Encrypted Relay for Packets) relays - dedicated servers that forward encrypted packets between nodes. This enables connectivity even in difficult network conditions.

Key technologies and protocols

- WireGuard - the underlying VPN UDP protocol used to encrypt packets between devices.
- DERP - a system of relay servers that mediates packets when a direct connection fails.
- NAT traversal / hole punching - a custom UDP protocol for punching through address translators.
- Noise IK / TLS - encryption and authentication protocols used to communicate with Tailscale servers.
- DNS and certificates - MagicDNS and automatic TLS certificates for resources in the tailnet.

Advantages of the solution

- End-to-end security - full encryption and privacy, private keys never leave the devices.
- No network configuration - zero-config operation, easy attachment of devices.
- Low latency and high performance - direct peer-to-peer connections.
- Flexibility - runs on multiple operating systems, integrates with Active Directory and cloud.
- Simple administration - through the web panel you can set ACL rules, define routes, monitor traffic.

Disadvantages and limitations

- Trust in the provider - traffic control and coordination requires connection to Tailscale servers.
- System requirements - need to install client on each device.
- Limits of the free plan - larger networks require a paid subscription.
- Performance on networks without direct connections - DERP relay can introduce latency.
- No VPN functionality for anonymisation - Tailscale is not a typical VPN for bypassing blockades or IP masking.

Practical applications

- Remote access to servers and devices - e.g. SSH, RDP, IP cameras without opening ports.
- Connecting multiple locations - connections between offices, clouds and dev/production environments.
- IoT and homelab - secure connections to home devices, Raspberry Pi etc.
- Zero Trust Networking - detailed access control and network segmentation.

Cloudflare Tunnel principle

Cloudflare Tunnel (formerly Argo Tunnel) enables secure sharing of server services (e.g. WWW, SSH, RDP) without the need for a public IP or open ports. A Cloudflared agent is run on the server, which initiates only outgoing, encrypted TLS connections to the Cloudflare infrastructure. User traffic from the Internet first goes to Cloudflare, which forwards it through a tunnel to the local server.

The tunnel can be bound to a domain name (e.g. CNAME) and Cloudflare provides encryption, filtering and protection against attacks. Connections are maintained over HTTP/2 or HTTP/3 (QUIC) for high performance and latency resilience.

Key technologies and protocols

- cloudflared - an agent on the server that maintains the connection to Cloudflare.
- TLS 1.3 - secures tunnelled data transmission.

- HTTP/2 and HTTP/3 (QUIC) - transport protocols supporting multithreading and low latency.
- Port 7844 / Cloudflare addresses - default port for communication with Cloudflare; only outbound access is needed.
- DNS and routing - the tunnel is bound to a DNS name, simplifying resource sharing.

Advantages of the solution

- No open ports - all connections are outbound.
- Protection against attacks - Cloudflare offers built-in DDoS protection and firewall.
- Global network - data is sent via the nearest possible route to the Cloudflare data centre.
- Simple configuration - quick installation and integration with Cloudflare Zero Trust panel.
- Broad application - supports many types of services (e.g. web servers, SSH, RDP).

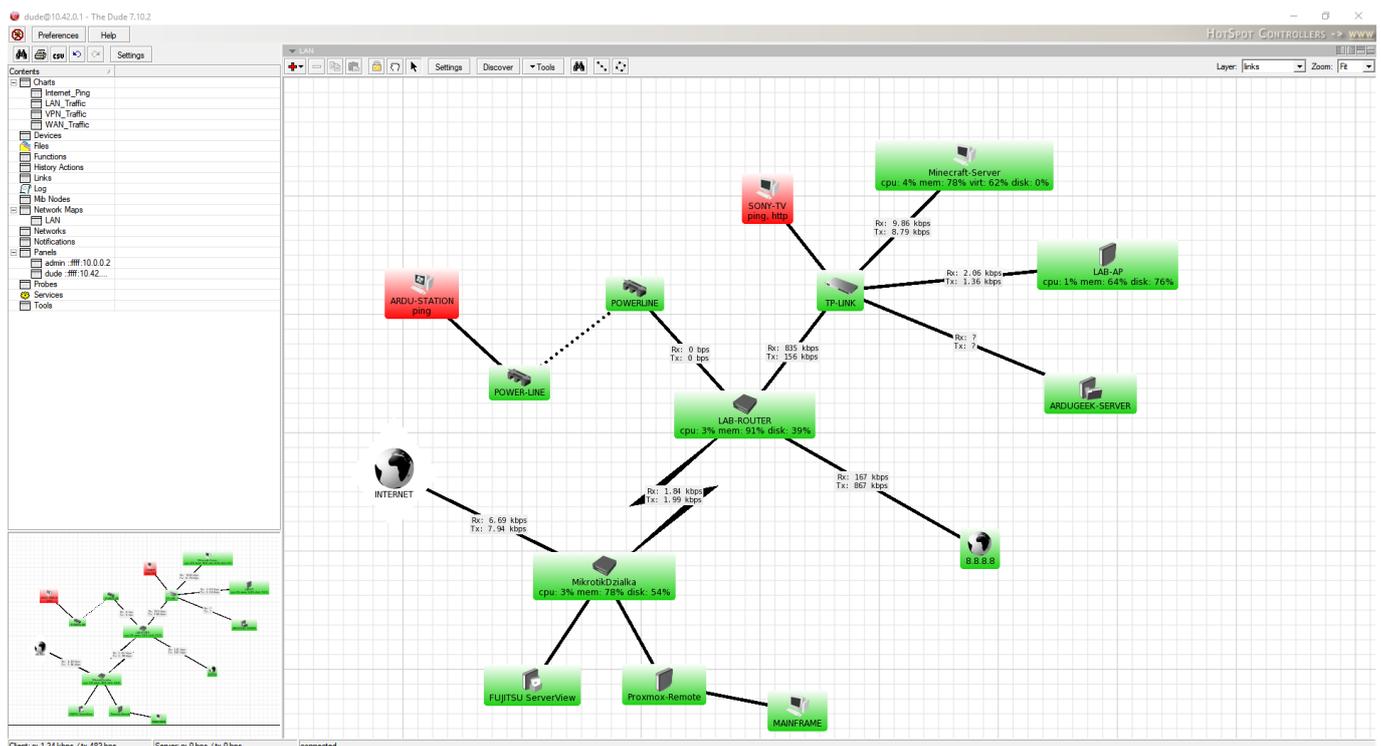
Disadvantages and limitations

- Dependency on Cloudflare - the tunnel only works through their infrastructure.
- Privacy - Cloudflare has technical access to decrypted HTTP requests.
- Cost - the free plan is limited, advanced features require a subscription.
- Latency - the extra hop over the Cloudflare network can introduce a small overhead.
- Outbound connection requirement - need access to port 7844 and Cloudflare addresses.

Practical applications

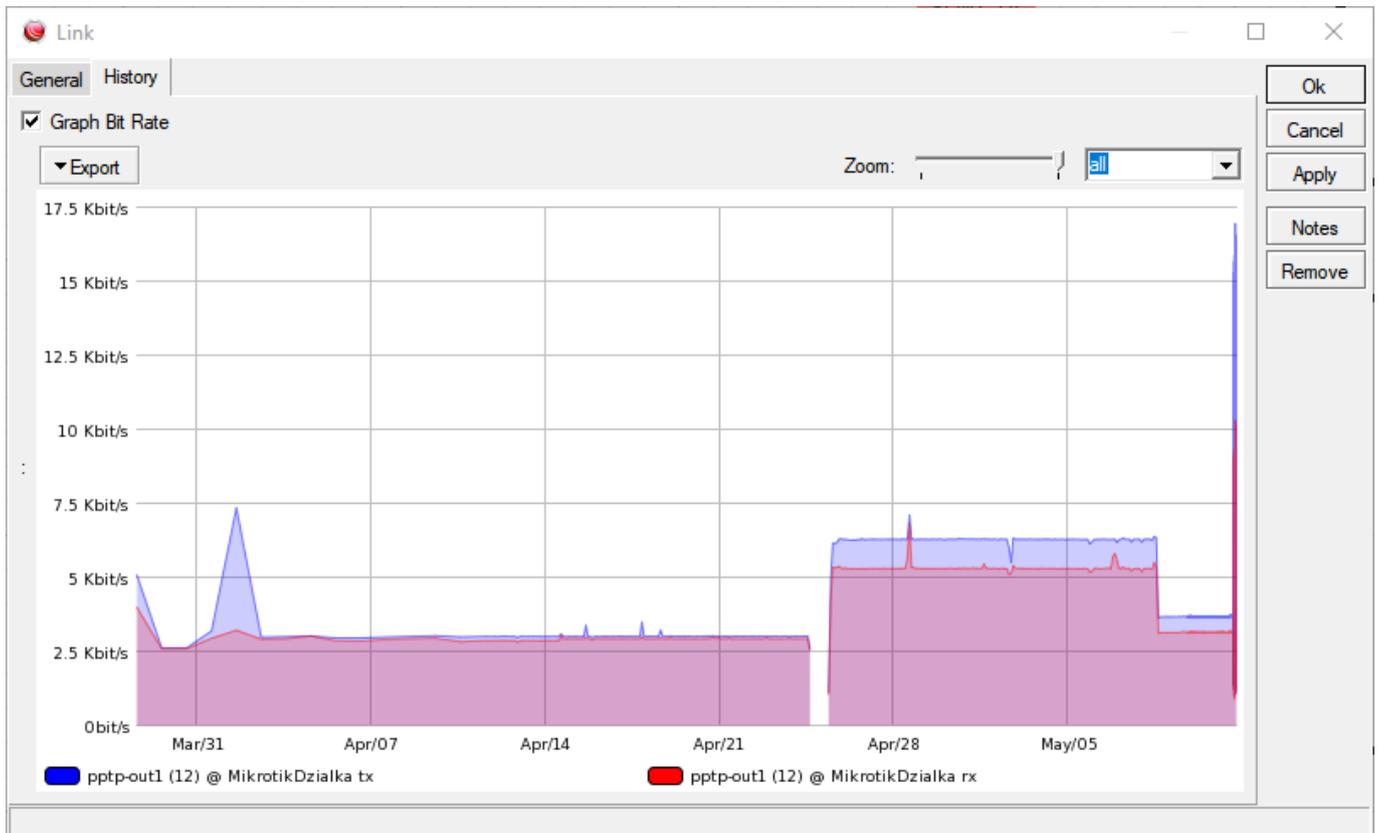
- Sharing web applications or APIs - without public IP and with access control.
- Remote access to desktops, servers or administration panels.
- Deployments in companies with limited Internet access or network behind NAT.
- Shared development environments - without external service exposure.

Monitoring my network



I do not have enough key services on my network to monitor them meticulously, but I do have, for

example, email notifications configured in cloudflare if one of the tunnels disconnects. As far as the rest of monitoring is concerned, I use the LAB-ROUTER on which I have the following service running [The Dude](#) it allows me to do a simple network map and simple monitoring of services such as Ping, HTTP, SSH etc.



I have simple traffic monitoring set up on all links which allows me to occasionally monitor for any anomalies.

Domains

As you can see in the earlier paragraphs I have two domains `ardugeek.ovh` and `ostrowski.net.pl`. I hasten to explain, at a time when I had few people I was teaching and giving training to I bought myself a domain to host my own site for experimentation and fun. As I got older and more experienced I came to the conclusion that it would be better to have an email and a domain with my own name on it, so I bought one too. Both are paid for for the next I don't remember how many years, so I still have them both, and I have adopted the convention that things that are more permanent such as this WIKI or the [moja website](#) will be under the domain `ostrowski.net.pl` and things that are experiments or just for fun I leave on the `ardugeek.ovh` domain.

Hence this genesis of two domains. One could say that one is IntraNet and the other is ExtraNet only that would be a bit of an incorrect use of these terms ☐.

Summary

In this article I have presented the design of my home network (some prefer to call it Home Lab



), which I use on a daily basis while listening to music on my smartphone or checking my notes on my WIKI. As you can see it is a mixture of many different services, programs and operating systems bundled together into one big (here everyone tells themselves what), this is the result of hosting various services over the years on various old laptops or recycled computers. This network configuration has been with me for a couple of years and I have the impression that this is the final version, from now on I only expect small changes but not a complete overhaul.

This network was and still is sometimes used to host the projects of my friends or family. At one time it was also used to host a number of websites for schools that I operated. Years of adding and removing technology has made it like a good old worn out sofa, when you sit on it you feel at home.