

VPN

virtual private network
VPN
Internet VPN (WAN)

- [n2n VPN](#)
- [FreeLAN](#)
- [Outline](#)
- [WireGuard](#)
- [Tinc](#)
- [Cloudflare Tunnel](#)
- [ngrok](#)
- [Chisel](#)

n2n VPN

Introduction

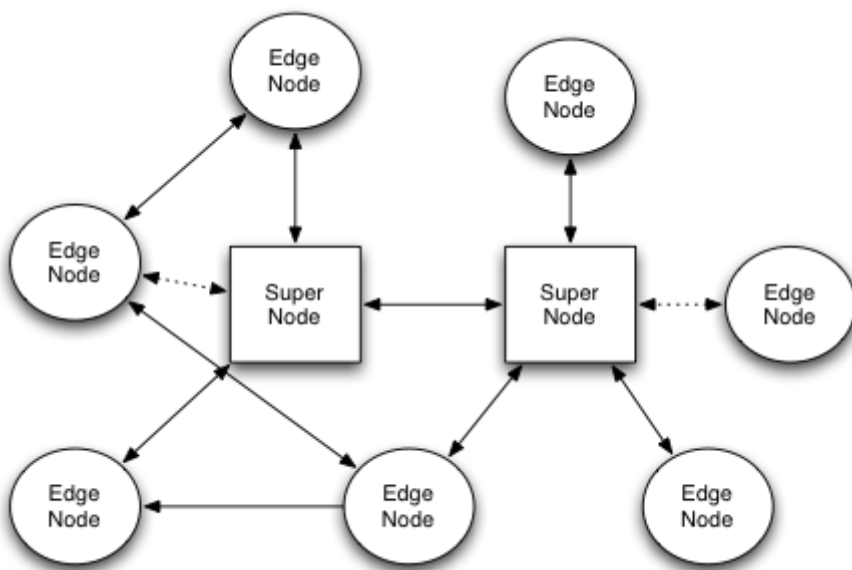
[n2n](#) is a layer-two **peer-to-peer** virtual private network (VPN) which allows users to exploit features typical of P2P applications at network instead of application level. This means that users can gain native IP visibility (e.g. two PCs belonging to the same n2n network can ping each other) and be reachable with the same network IP address regardless of the network where they currently belong. In a nutshell, as OpenVPN moved SSL from application (e.g. used to implement the https protocol) to network protocol, n2n moves P2P from application to network level.

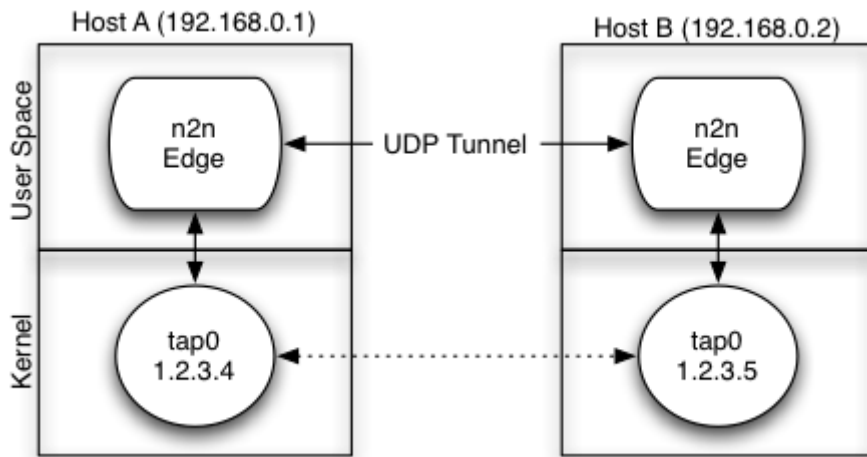
In order to start using n2n, two elements are required:

- A *supernode*: it allows edge nodes to announce and discover other nodes. It must have a port publicly accessible on internet.
- *edge nodes*: the nodes which will be a part of the virtual networks

A virtual network shared between multiple edge nodes in n2n is called a *community*. A single supernode can relay multiple communities and a single computer can be part of multiple communities at the same time. An encryption key can be used by the edge nodes to encrypt the packets within their community.

n2n tries to establish a direct peer-to-peer connection via udp between the edge nodes when possible. When this is not possible (usually due to special NAT devices), the supernode is also used to relay the packets.





Installation

Download: <https://github.com/ntop/n2n>

Ubuntu/Debian

```
sudo dpkg -i n2n_3.0.0-1038_amd64.deb
sudo apt install -f
```

Usage

One-liner command in foreground.

```
# On Linux, change the community, encrypt key and tun IP to your own
# edge -c <community> -k <encrypt key> -a <tun IP address> -l <supernode host:port> -f
sudo edge -c my-community -k my-secret -a 10.9.9.10 -l n2n.lucktu.com:10090 -f
```

Run as service in background.

```
# Generate the config file
sudo cp /etc/n2n/edge.conf.sample /etc/n2n/edge.conf

# Start the edge
sudo systemctl start edge
sudo systemctl enable edge
```

Supernode

```
sudo supernode -p 10090
```

????

- <https://github.com/ntop/n2n>
- [n2n for Windows](#)
- [n2n GUI for Win](#)
- [OpenVPN Download](#)
- [Public Supernodes](#)
- [ZeroTier](#) - P2P VPN Service
- [FRP](#) - Fast Reverse Proxy
 - [Video] [?????FRP????????????????](#)

FreeLAN

Introduction

[Freelan](#) is a free, open-source, multi-platform, peer-to-peer VPN software that abstracts a LAN over the Internet. It works on Windows, Linux and Mac OSX.

Whether you want to connect the computers of your family, play an old LAN-only game with your friends, or give a privileged access to your private network to your collaborators, freelan will do the job perfectly.

[FreeLAN](#) is free all-around VPN open-source software for Windows, Linux, and macOS that can be used to create three types of VPN:

- Client-server
- Peer-to-peer
- Hybrid that includes the two types mentioned above.

Installation

Download: <https://www.freelan.org/download.html>

Ubuntu/Debian

```
sudo apt update
sudo apt install freelan
```

Windows

1. [?? Visual C++?](#)
2. [?? freelan-2.2.0-amd64-install.exe?](#)

Usage

One-liner command in foreground.

```
# On PC#1, which is correctly configured to allow access to the 12000/UDP from the internet.
# By default, the listen port is 12000/UDP and the tun_tap.ip_address is 9.0.0.1
freelan --security.passphrase "my_secret"
```

On PC#2

```
freelan --security.passphrase "my_secret" --tap_adapter.ipv4_address_prefix_length 9.0.0.2/24 --fscp.contact  
<IP-to-PC1>:12000
```

Outline

??

[Outline](#) ?????????????? VPN???????????????????? Outline ???? VPN
????????????????????????????????

????

- <https://getoutline.org/zh-TW>
- <https://github.com/Jigsaw-Code>
- [??OUTLINE](#)
- [Outline VPN????? VPN ?????](#)
- <https://outline.community/>

For ARM CPU

- [Outline Server with ARM64 support](#)

WireGuard

Tutorials

Installation

- [wireguard-install](#)

Algo VPN

- [Algo VPN](#) is a set of Ansible scripts that simplify the setup of a personal WireGuard and IPsec VPN.

NetBird

- [NetBird](#) combines a configuration-free peer-to-peer private network and a centralized access control system in a single open-source platform
- [Video] [Netbird - an Open Source, Self Hosted Wireguard based VPN system. Server GUI and client setup ease](#)

Tinc

Tinc is free and open-source VPN software that can be used to create mesh VPN networks. It is a small and powerful peer-to-peer VPN daemon that can be installed on multiple platforms. Tinc uses encryptions and tunneling for creating a secure private network between multiple hosts.

Requirement:

- ?? node ??? Public IP

Tutorials

- [How to Set Up Peer-to-Peer VPN with Tinc on Ubuntu 22.04](#)

Cloudflare Tunnel

Cloudflare Tunnel ?????????????? Argo Tunnel

```

???????????????????????????????????????????????????????????????????????? Cloudflare
???????????????? Tunnel ????????? Router ????????? Port ?????????

```

??? Firewall ?????????? port ???????????? NAT ??????

- [Cloudflare Tunnel](#)
- [Cloudflare Docs: Cloudflare Tunnel](#)
- [?? Cloudflare Tunnel ????????????](#)
- [Video] [You Need to Learn This! Cloudflare Tunnel Easy Tutorial](#)

Twingate - 5 users for free

- [Docs: Twingate](#)
- [Video] [the END of VPNs?](#)

ngrok

ngrok - put localhost on the internet

?????

- ????????????????
- ??? dev/test ?????????? webhook????????????????????????????????

Pyngrok

ngrok SDK for Python

- [pyngrok - a Python wrapper for ngrok — pyngrok 7.1.6 documentation](#)
- GitHub: <https://github.com/alexdlaird/pyngrok>

URLs

- [ngrok | Unified Application Delivery Platform for Developers](#)
- [?? ngrok ?????????? localhost ??? - MyApollo](#)

?????

- [Localtunnel ~ Expose yourself to the world](#)
- [localhost.run | localhost.run](#)

Chisel

Chisel is a fast TCP/UDP tunnel, transported over HTTP, secured via SSH. Single executable including both client and server. Written in Go (golang). Chisel is mainly useful for passing through firewalls, though it can also be used to provide a secure endpoint into your network.

- GitHub: <https://github.com/jpillora/chisel>
- [Chisel ???? TCP ?? - iT ????::???????????? IT ???? \(ithome.com.tw\)](http://ithome.com.tw)
- [Chisel: Secure TCP/UDP Tunneling for Modern Networks - Deniz Halil](#)