# DynDNS and IPv6

I found out, that IPv6 is really nice - even if you are behind some firewall or router - as long as you can make them pass the packets to your host. By default most home office routers would block the packages, as otherwise every host in your local network would be completely open to the internet. This - in fact - makes it really nice to manage, as you wont need tricks to reach your host directly from out of the internet as IPv4 needs them.

## Difference between IPv4 and IPv6

Also IPv6 is quite old (released in 1998), it is still not very common as its setup is a bit different from the simple IPv4.

IPv4 works with NAT, which means, that the Firewall/Modem of your Network is the central point, which can be reached from the Internet. The Firewall then gets configured to accept connections on port- numbers an redirect them to hosts on the local net. The Internet only sees your Modems IPv4- Nummber and the Host on your network too. There is no knowledge of your internals Host-IP-Adress from the Internet perspektive.

IPv6 instead, has no NAT. The Modem gets an Adress assigned from the Internet Provider (called GUA) and the first half of it is called the prefix (the four Numbers in front of the first four colons). All Adresses to the right of that Adress are free to choose and are useable by Devices in the internal Network.Mostly the Devices are getting the Adresses via DHCP from the Modem which is mostly configured to work out of the Box, or they will choose their own IPv6.

These Device- Adresses are globally unique (GUAs) and are routable over the Internet. So Devices can be reached directly under that Adress - no matter which Modem is in between.

So basically every Host in the local Network can be reached from the Internet with ist GUA IPv6 Adress. Its the Modems Firewall, which will prevent exposing all Hosts of the local network directly to the Internet. Thus, the only thing you have to do, is to open the Ports for some Device, which the Firewall will allow to pass from the internet to the local network directly.

## Problems with Modems

I experienced, that opening the ports a some Modems, is basically possible, but still no connection could be made.

### Problematic Brands

For example, the German Telekom does have a modem / router called „Speedport“, which is generally not allowing IPv6 from the internet to pass to the local net (while IPv4 with NAT works as expected).

So - if you did the settings at your Firewall and still you cannot curl some Adress, maybe you have the wrong modem.

### Settings

Also check if your modem has some feature called „rebind protection“. If so, you need to add ALL full names to the list of allowed services.

e.g. add the fqdn like:

```
cname.domainname.dynv6.net
```

How the rebind protection work: DNS queries to your Router won't return the IP of the service. So if you experience, that `dig to 8.8.8.8` will work, but `nslookup` does not - check the rebind protection!

## Modem Setup

For IPv4 you would need to setup the Modem to have the DynDNS Record updated - as the Modem knows its own IP best and is the central Point of action.

Not so with IPv6. The Modems IPv6- Adress is irrelevant for your internal Hosts and for the Internet. As the IPv6 adress is assigned to the Device at connection establishment, which could be anywhere, it makes more sense, to have the device itself update the dnydns- record to its own adress.

So you can skip setting up the dyndns- Account in your Modem/Router.

### Firewall

There is one thing to do in your router: open the Device and the Port to be accessable from the internet. This is nearby the same as it is done at IPv4. After that, genereally the host shpuld be reachable.

## Security/privacy Extensions

By default, your IPv6 Adress will contain the MAC of your network card, which is an unique hardware- identifier of the chip. This has the advantage, that this part of your adress is fixed in the internet an will not change, so you can always reach your device anywhere knowing this adress part - even on mobile devices that may change the adress according to the connected network.

The opposite is, that your device can always be identified by that adress for all times - making all network traffic attached to the device directly belonging to it. So someone analysing the traffic of that adress would know exactly when and what has been done with that device. This is very poor as for privacy.

To avoid this, your device can generate an id that will change from time to time, making it impossible to identify the device with that adress.

To turn this on, you can add a line to `/etc/sysctl.conf` like:

```
sudo sysctl net.ipv6.conf.wlan0.use_tempaddr=2
```

Replace wlan0 with the name of your Device.

After that, check if your device has an temporary adress with `ip addr`

*** Not finished - i have not found any docs at ddclient, how to get the interface temporary adress working, made up https://github.com/ddclient/ddclient/issues/651** * For the time beeing, i will be fine using the non- private adress.

## DynDNS Provider

Next Step to choose is a provider for dyndns. There are many dyndns- providers out there - even free of charge. E.g. https://dynv6.com which seems to work fine. Registration there is done quite fast and no setup needs to be done at the providers interface.

The opposite of those free registries is, that you can only use a subdomain of the Top-Level domain they offer, which makes your Domain Name fixed at the end. For me personally, i have bought a domain on my own in the tld of my country (costs about 15 Euro in one Year), which i can now use.

## Register Subdomains

After you logged into your DynDNS Provider, enable dynDNS for your IPs and add subdomains - each one for one service. If you want to acces your Portainer you

created when setting up docker, e.g. use a Subdomain called

portainer.domain.tld

Make sure, that DynDNS is selected for that record again!

## DDclient

The Task to update the dyndns- entry to point to the right host can be done best on the host itself as explained before. The Host may detect interface- changes of the Adress and will push out automagically using some client.

Hint for old SuSE: DDClient can be used in the local server/host to assign some DNS- Record to an IP. It turned out, that the **DDclient V3.8.3 of OpenSuSE Leap 15.5** was not able to work with IPv6 in the Version coming from the main Repositories. So add the Repository `https://download.opensuse.org/repositories/network/${releasever}/` with yast and update to the Packages of that Repository first, so you get **DDclient V3.11.3**, which is able to deal with ipv6.

After upgrading to Leap 15.6, this should be gone.

The config is in `/etc/ddclient/ddclient.conf` and it could contain e.g. those lines:

```
# Globals
daemon=300
ssl=yes
syslog=yes
mail-failure=root

# IP-Specific
usev6=ifv6,
if=eth0,
protocol=dyndns2,
server=dyndns.strato.com/nic/update,
login='domain.tld', password='blabla',
nextcloud.domain.tld,ipa.domain.tld

# If you want ipv4 too you may use:
usev4=cmdv4, cmdv4=/etc/ddclient/myipv4address.sh
```

```
protocol=dyndns2,
server=dyndns.strato.com/nic/update,
login='domain.tld',
password='blabla',
nextcloud.domain.tld,ipa.domain.tld
```

Notes:

- Not using SSL will not work
- DDClients Config is a bit strange to understand. My example is **one server** in the view of ddclient, which is why there are commas seperating the options for that host.
- You can also specify each option without comma, but then the setting will change the **defaults** of ddclient

After that, you may use a script for determining the right ipv4- adress. This is while NAT of IPV4 needs the adress of your router and not you servers adress.

One Example would be this file /etc/ddclient/myipv4address.sh

```
#!/bin/bash

curl -s4 http://ifconfig.me/ip
```

Or - if you are using Fritzbox, then you may use phyton:

```
#!/bin/bash
FULLSTR=$(/etc/ddclient/get_ipaddress.py)
SEARCH='"NewExternalIPAddress":'
S2='",'
P1=${FULLSTR#*$SEARCH}
P2=${P1%%$S2*}
IPADDR=${P2:2}
echo -n ${IPADDR}
```

And File /etc/ddclient/get_ipaddress.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# Quelle: https://github.com/ran-sama/fritzbox-soap-json
```

```
# Vorbereitung:
# - Paket python3-xmltodict installieren
# - Benutzer, Passwort

import requests, xmltodict, json, re
from requests.auth import HTTPDigestAuth

def main():
    # define your IP and credentials in you fritzbox first
    username = "fritzboxuser"
    password = "passwordfritzboxuser"

    # what we want to access
    req_endpoint = '/upnp/control/wanpppconn1'
    service = 'urn:dslforum-org:service:WANPPPConnection:1'
    action = 'GetInfo'

    # form-autofill for python users
    soapaction = service + '#' + action
    raw_envelope = re.sub(r"\s +", "",
        """<?xml version="1.0" encoding="utf-8"?><s:Envelope s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body><u:{action}
xmlns:u="{service}"></u:{action}></s:Body></s:Envelope>""")

    # send the authenticated soap request
    auth = HTTPDigestAuth(username, password)
    device = "http://fritz.box:49000" + req_endpoint
    headers = {'soapaction': soapaction, 'content-type': 'text/xml','charset': 'utf-8'}
    envelope = raw_envelope.format(service=service, action=action)
    encoded = envelope.encode("utf-8")
    boxdata = requests.post(url=device, data=encoded, headers=headers, auth=auth).content.decode('utf-8')

    # XML to dict, remove outer nesting, pretty print JSON
    data_dict = xmltodict.parse(boxdata)
    response_tag = 'u:' + action + 'Response'
    data_dict = data_dict['s:Envelope']['s:Body'][response_tag]
    json_data = json.dumps(data_dict, indent=4)
    print(json_data)
```

```
main()
```

## Debugging

If something is not working, execute ddclient that way:

```
ddclient -daemon=0 -debug -verbose -noquiet
```
The programm can be executed directly.

## Enable the service

If your IP has been sucessfully updated, enable the service:

```
systemctl enable ddclient
```

## Check the Connection

you may now have the IPv6 of your Device published at some adress and check this by digging at googles DNS for that IPV6:

```
pcserver2023:~ # dig @8.8.8.8 ipa.obel1x.de AAAA

; <<>> DiG 9.18.33 <<>> @8.8.8.8 ipa.obel1x.de AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50334
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ipa.obel1x.de.                 IN      AAAA

;; ANSWER SECTION:
```

```
ipa.obel1x.de.          60      IN      AAAA    2a00:1f:f8c1:6d01:468a:5bff:fe9f:6415

;; Query time: 44 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Sat Feb 08 12:49:12 CET 2025
;; MSG SIZE  rcvd: 70
```

Thats it, you shold be able to get some connection. Mind, that IPv6- Adresses in URLs are written in brakets to have ports seperated:

```
protocol://[ipv6adress]:port
```