Freifunk Troisdorf Gateway Doku

Release 0.1 beta

Freifunk-Troisdorf

May 03, 2018

Contents

In dieser Doku beschreiben wir unser Gluon/Linux Gateway Setup. Diese Daten stammen aus unserem Netz in Troisdorf, weshalb auch unsere IP Netze in den Config dateien zu finden sind.

Dieses Setup ist in mehreren Monaten entstanden, und kann dennoch noch einige Fehler enthalten. Wir versuchen unsere Gedankengänge hier so gut es geht zu umschreiben.

CHAPTER 1

Intro

1.1 Voraussetzungen

- voller Root Zugriff für alle Admins
- öffentliche IPv4 Adresse
- öffentliche IPv6 Adresse
- **voller Kernelzugriff**
  - Linux direkt installiert,
  - oder als KVM-Instanz (OpenVZ o.ä. nicht möglich)
- Hier genutzt: Debian 8

1.1.1 Betrieb und Zugang

Unsere Server werden momentan von 2 Personen Privat betriben. In unserem Setup ist immer nur 1 Server Aktiver Supernode, um unter anderem den Inter Gateway Traffic zu sparen. Dies funktioniert mit L2TP sehr gut, da die vorhandene Performance mit 1 Server ausreichend ist.

Wir installieren unsere Gateways per Ansible, was uns ein Identisches Setup per Server ermöglicht und uns neue Gateways innerhalb weniger Minuten ausrollen lässt. Die Ansible Config findet Ihr hier: https://github.com/Freifunk-Troisdorf/ansible.fftdf.supernode

1.1.2 Funktionen & dafür benötigte Software

Momentan kommen auf allen Gateways Debian 8 (jessie) zum Einsatz.

Für viele der Gate-Funktionen wird weitere Software benötigt, die alle aus den Debian Repositories nachinstalliert werden können.

See also:

- *Pakete*

Verbindung der Server und Nodes

Die Nodes bauen ihr Freifunk mit dem Mesh-Protokoll *B.A.T.M.A.N* selbständig auf.

Da das Netz aber noch nicht dicht genug ist, dass alle Nodes sich über ihre Funk-Interfaces erreichen können, helfen wir über eine VPN-Verbindung über das Internet nach, um einzelne Wolken zu überbrücken.

So wie über ihre Funk-Interfaces, sprechen die Nodes das *B.A.T.M.A.N* Protokoll auch über ihre VPN-Verbindungen.

Die Gateways sind die VPN Server für unsere Nodes. Die Nodes bauen alle einen L2TP Tunnel zu diesen auf.

Alle Gateways verbinden sich für das VPN auch vollständig untereinander. So entsteht eine Multi-Stern-Architektur. Alle Sternmittelpunkte sind untereinander voll vermascht.

Auf den Gateways wird hierzu **L2TP** sowie das Kernel Modul **batman_adv** benötigt.

See also:

- *Pakete*
- l2tp

DHCP für die Clients

Nodes, Clients und Gateways sind über ein Layer-2-Netz miteinander verbunden (*B.A.T.M.A.N*, s. o.).

Die Gateways halten auch als DHCP-Server für die Clients her.

Verfügbare IPv4-Ranges, aus denen der DHCP-Server vergeben darf, müssen innerhalb bzw. mit der Admin-Gruppe abgestimmt werden.

Hierfür wird **isc-dhcp-server** genutzt. Für das vorbereitende Ausrollen von IPv6 Adressen benötigen wir hier auch **radvd**.

See also:

- dhcp
- radvd

Übergang ins restliche Internet

Die Verbindung ins Internet wird über den Freifunk Rheinland e.V. realisiert. Wir verbinden uns hierzu per **GRE** in das Backbone des FFRL und bekommen von hier auch Public IPv4 und IPv6 Adressen.

See also:

- interfaces
- internetexit

Datenschutz auf dem Gateway

Unsere Gateways loggen keinen Traffic! Es werden lediglich Statistiken für die Bewertung der Auslastung unserer Gateways gespeichert.

Alles was existiert sind die zur Laufzeit benötigten Verbindungsdaten. DHCP-Leases, Batman Protokolldaten und die ARP-Tabelle.

Diese werden nur im Arbeitsspeicher vorgehalten, ist das Gateway aus (z.B. die Herren in Grün nehmen den Server mit), sind diese i.d.R. weg.

See also:

- logging

1.2 Netzplan

Freifunk Troisdorf ist ein IP Netz aus dem IPv4 Bereich `10.188.0.0/16`.

Wir haben unser IP Netz in Segmente eingeteilt um eine bessere Überischt zu haben.

1.2.1 IPv4

Wir haben für unserer Freifunk folgende Netze zugewiesen bekommen:

- Troisdorf - `10.188.0.0/16` = **188**

Datenpakete aus diesem Adressbereich werden innerhalb des Freifunks vermittelt, im großen weiten Internet aber nicht geroutet (Hintergrundinformationen dazu gibt es hier).

Unser großes `10.188.0.0/16` (mit 65536 Adressen) teilen wir uns ein wenig ein:

`10.188.0.0/16` wird nicht komplett genutzt, um in Zukunft noch was auf Halde zu haben. Wachsen ist immer einfacher als schrumpfen.

Netz	(bis)	Verwendung	verteilt durch	status
`10.188.1.0/24`	`10.188.1.255`	Services	fix	in Betrieb
`10.188.2.0/24`	`10.188.2.255`	Static IP Leases	Github/DHCP	Geblockt
`10.188.100.1`	`10.188.103.255`	Client DHCP-Range	troisdorf1	frei
`10.188.104.1`	`10.188.107.255`	Client DHCP-Range	troisodrf2	frei
`10.188.108.1`	`10.188.111.255`	Client DHCP-Range	troisdorf3	frei
`10.188.112.1`	`10.188.115.255`	Client DHCP-Range	troisdorf4	fre
`10.188.116.1`	`10.188.119.255`	Client DHCP-Range	troisodrf5	in Betrieb
`10.188.120.1`	`10.188.124.255`	Client DHCP-Range	troisdorf6	in Betrieb
`10.188.125.1`	`10.188.128.255`	Client DHCP-Range	troisdorf7	frei
`10.188.255.0/24`	`10.188.255.255`	Gateway IPs	fix	Geblockt

NEU:

Wir teilen das /16 in 8 /19 Netze ein.

Sub-Domäne 1 - Troisdorf

IPv4: `10.188.0.0/19`

Netz	(bis)	Verwendung	verteilt durch	status
10.188.0.0/21	10.188.7.255	Netz Intern	fix	in Betrieb
10.188.8.0/21	10.188.15.255	Client DHCP-Range	•	in Betrieb
10.188.16.0/21	10.188.23.255	Client DHCP-Range	•	backup
10.188.24.0/21	10.188.31.255	frei	•	frei

IPv6: 2a03:2260:121:4000::/64 wird per Radvd announced

Sub-Domäne 2 - Troisdorf City IPv4: 10.188.32.0/19

Netz	(bis)	Verwendung	verteilt durch	status
10.188.32.0/18	10.188.39.255	Freie Verwenung	fix	in Betrieb
10.188.40.0/18	10.188.47.255	Client DHCP-Range	•	Geblockt
10.188.48.0/18	10.188.55.255	Client DHCP-Range	•	frei
10.188.56.0/18	10.188.63.255	Netz	•	frei

IPv6: 2a03:2260:121:5000::/64 wird per Radvd announced

Sub-Domäne 3 - Flüchtlinge IPv4: 10.188.64.0/19

Netz	(bis)	Verwendung	verteilt durch	status
10.188.64.0/18	10.188.71.255	Freie Verwendung	fix	frei
10.188.72.0/18	10.188.79.255	Client DHCP-Range	•	frei
10.188.80.0/18	10.188.87.255	Client DHCP-Range	•	frei
10.188.88.0/18	10.188.95.255	Client DHCP-Range	•	frei

IPv6: 2a03:2260:121:6000::/64 wird per Radvd announced

Subdomäne 4 - Events IPv4: ‘‘10.188.96.0/19

Netz	(bis)	Verwendung	verteilt durch	status
10.188.96.0/18	10.188.103.255	Freie Verwendung	fix	frei
10.188.104.0/18	10.188.111.255	Client DHCP-Range	•	frei
10.188.112.0/18	10.188.119.255	Client DHCP-Range	•	frei
10.188.120.0/18	10.188.127.255	Client DHCP-Range	•	frei

IPv6: 2a03:2260:121:7000::/64 wird per Radvd announced

1.2.2 IPv6

Wir nutzen das uns vom FFRL zugewiesene Public IPv6 Netz 2a03:2260:121::/48

IPv6 Subnetze haben immer eine Prefix-Länge von *64 Bit*. Durch das /48 Subnetz stehen uns also 2^16 = 65536 /64 IPv6 Subnetze zur Verfügung.

1.2.3 Interface Bezeichnung

Wir vergeben unsere Interface-Bezeichnungen einheitlich!

	eth0	Mesh - Bridge (Nodes)	B.A.T.M.A.N	Inter Gateway VPN	Exit VPN
		br-[SUB Name tdf, inn, flu]	bat[SUB NAME]	l2tp-*	gre-bb-*

1.2.4 Namenskonvention

Unsere Gateways sind durchnummeriert. Angefangen bei **troisdorf0** bis zu **troisdorf9**.

1.2.5 Next Node Adressen

Die Next Node Adressen sind dafür da, um sich im Fehler- oder Troubleshootingfall mit einem Freifunk Knoten zu verbinden.

Diese Adressen sind auf jedem Knoten gleich. Der Freifunker muss sich nur diese Adresse(n) merken und seine Netzwerkkarte für ein Subnetz konfigurieren, in dem diese Adresse(n) liegt um auf seinen Knoten zuzugreifen.

Wir nutzen dazu die jeweils niedrigsten Adressen

- **Troisdorf:**
  - IPv4: 10.188.0.1
  - IPv6: 2a03:2260:121::1

1.2.6 Gateway-Schema

Bevor wir ein Gateway aufsetzen definieren wir einen Namen, dessen Nummer auch gleichzeitig in vielen Scripten genutzt wird.

Mit den uns zugewiesenen Netznummern sowie der Gateway-Nummer und dem Gateway-Namen werden alle benötigten Informationen abgeleitet:

- **IPv4**
  - Für Gateways wird das Subnetz `10.188.255.0/24` verwendet. Die Adressen sind bereits definiert. Beispiel troisdorf1: `10.188.255.1`
- **MAC-Adresse**
  - Privates Prefix (`0a2:8c:ae:6f:f6:**`) + Gatewaynummer
  - **Beispiele:**
    - 10.188.255.1 -> `a2:8c:ae:6f:f6:01`
    - 10.188.255.2 -> `a2:8c:ae:6f:f6:02`
- **IPv6**
  - Range-Prefix (`2a03:2260:121::255:`) + Gatewaynummer
  - **Beispiele:**
    - troisdorf1 -> `2a03:2260:121::255:1/64`
    - troisodrf2 -> `2a03:2260:121::255:2/64`
- **DNS**
  - `troisdorf[1-9].freifunk-troisorf.de` -> A- + AAAA-Record
  - `[1-9].fftdf.de` -> CNAME auf s.o.
  - Reverse DNS Eintrag korrekt setzen für Haupt DNS Namen: `troisdorf[1-9].freifunk-mwu.de`

1.2.7 Beispiel

Gateway: **troisdorf5** - Nummer: **5**

troisdorf5	
IPv4	`10.188.255.5`
IPv6	`2a03:2260:121:5000::5`
MAC	`a2:8c:ae:6f:f6:05`
DNS1	`troisdorf5.freifunk-troisdorf.de`
DNS2	`5.fftdf.de`

1.3 Gluon

Gluon entstand in Lübeck aus dem Bedarf heraus, einheitliche OpenWRT-Images für Freifunk-Nutzer herauszugeben, die bereits vorkonfiguriert und mit allen Packages versehen waren.

Da es nicht nur in Lübeck Freifunk gibt, wurden alle Elemente zusammen getragen, modularisiert, und für alle Freifunk-Communities veröffentlicht. *Vielen Dank!*

Es ist ein Aufsatz, der den normalen OpenWRT Source Code hernimmt, patcht, und anhand der Konfiguration alles zurechtrückt um den OpenWRT-Bauprozess für einen Satz unterstützter Router-Modelle zu starten. Dabei können dem OpenWRT noch weitere Pakete untergeschoben werden.

Heraus fallen fertige Images, vorkonfiguriert, nur noch flashen ist nötig. Bonus: Die Images werden noch fein mit elliptischen Kurven signiert, um ein automatisches Updaten zu ermöglichen (Autoupdater ist ein Standard OpenWRT-Paket für Gluon)

1.3.1 Hintergrund zu Gluon

Ein bisschen zu den Anfängen von Gluon und viel mehr zum technischen Hintergrund kann man unter folgenden Links erfahren:

- http://luebeck.freifunk.net/wiki/gluon (http://luebeck.freifunk.net/wiki/Firmware)
- http://luebeck.freifunk.net/wiki/fastd-schlusselverwaltung
- http://nilsschneider.net/2012/12/24/openwrt-autoupdater.html
- http://nilsschneider.net/2013/02/17/fastd-tutorial.html

1.3.2 Links & Dokumentation

- Gluon Code: https://github.com/freifunk-gluon/gluon
- Gluon Pakete:
  - https://github.com/freifunk-gluon/gluon/tree/master/package
  - https://github.com/freifunk-gluon/packages
- Gluon Konfiguration (site.conf) https://github.com/freifunk-gluon/gluon/tree/master/docs/site-example
- Dokumentation: http://gluon.readthedocs.org/
- Github Wiki: https://github.com/freifunk-gluon/gluon/wiki
- fastd: http://fastd.readthedocs.org/
- Doku aus Hamburg: https://wiki.freifunk.net/Freifunk_Hamburg/Firmware#Gluon

CHAPTER 2

Konfiguration

2.1 Grundkonfiguration

2.1.1 Hostname

Den Hostnamen setzen (ohne Domain):

```
echo "hostname" > /etc/hostname
```

/etc/hosts (am Beispiel von troisdorf5):

```
127.0.0.1   localhost
127.0.1.1   troisdorf5.freifunk-troisdorf.de    troisdorf5

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
5.9.76.198 troisdorf5.freifunk-troisdorf.de troisdorf5
2a01:4f8:161:62a9::5 troisdorf5.freifunk-troisdorf.de troisdorf5
```

2.1.2 Routing Tabellen

/etc/iproute2/rt_tables:

```
#
# reserved values
#
255 local
254 main
253 default
0   unspec
```

```
#
# local
#
#1  inr.ruhep
42 ffrl
```

2.1.3 Pakete

Pakete aus den Standard-Repos installieren:

```
apt-get install -y git make gcc build-essential pkg-config libgps-dev libnl-3-dev␣
↪libjansson-dev isc-dhcp-server collectd libcap-dev iproute libnetfilter-conntrack3␣
↪python-dev libevent-dev ebtables python-virtualenv iptables-persistent iftop screen␣
↪bridge-utils tcpdump bind9 radvd curl htop psmisc dnsutils ntp
```

2.1.4 NTP

Da die Kisten recht viel mit Crypto machen, ist es von Vorteil eine halbwegs genaue Uhrzeit parat zu haben.

Die `/etc/ntp.conf` bleibt nahezu unverändert:

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

# Specify one or more NTP servers.
server 0.de.pool.ntp.org
server 1.de.pool.ntp.org
server 2.de.pool.ntp.org
server 3.de.pool.ntp.org

# Use Ubuntu's ntp server as a fallback.
server ntp.ubuntu.com

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
```

CHAPTER 3

Betrieb

CHAPTER 4

Firmware

4.1 v1.0

See also:

Github site.conf

- Erstes Gluon Release nach dem Netsplit

4.2 v1.0.2

See also:

Github site.conf

- Anpassung der IPv6 Adressen in den neuen netzen

4.3 v1.0.3

See also:

Github site.conf

Autoupdater Bugfix

4.4 v1.0.5

See also:

Github site.conf

- Bugfixes

4.5 v1.0.7

See also:

Github site.conf

- Fehlerbehebungen im Reboot und Wifi Restart Script

4.6 v2017.1.1-1

See also:

Github site.conf

- Änderrung Versionsnummern auf [GLUON-VERSION]-[BUILD-NUMMER]
- Gluon Release 2017.1.1

4.7 v2017.1.2-1

See also:

Github site.conf

- Gluon Release 2017.1.2

4.8 v2017.1.3-1

See also:

Github site.conf

- Gluon Release 2017.1.3

4.9 v2017.1.4-1

- Täglicher reboot in Wöchentlichen Reboot geändert
- Täglicher reboot wenn keine Clients auf dem Node sind um 4 Uhr
- ATK10 Images werden wieder gebaut
- Gluon Release 2017.1.4

See also:

Github site.conf

- Update auf Gluon Master wegen Wifi Problemen

4.10 v2017.1.5-1

- Update auf Gluon 2017.1.5
- Hardware Support für TP-Link TL-WR1043N v5, Ubiquiti EdgeRouter-X, Ubiquiti EdgeRouter-X SFP

See also:

Github site.conf

See also:

Gluon Release Notes

4.11 v2017.1.6-1

- Update auf Gluon 2017.1.6

Rollout wegen Bug abgebrochen

See also:

Github site.conf

See also:

Gluon Release Notes

4.12 v2017.1.7-1

- Update auf Gluon 2017.1.7
- Bugfix für Bootloader-Bug in 2017.1.6-1

See also:

Github site.conf

See also:

Gluon Release Notes

CHAPTER 5

Mitwirken

Du hast eine Stelle entdeckt, die Schreibfehler enthält, was unsauber oder falsch erklärt, nicht existiert aber sollte, oder veraltet ist?

Gerne nehmen wir sinnvolle Vorschläge und Ergänzungen mit auf, über den Issue-Tracker (mit dem Label ‘Doku’) oder am besten gleich per Pull Request.