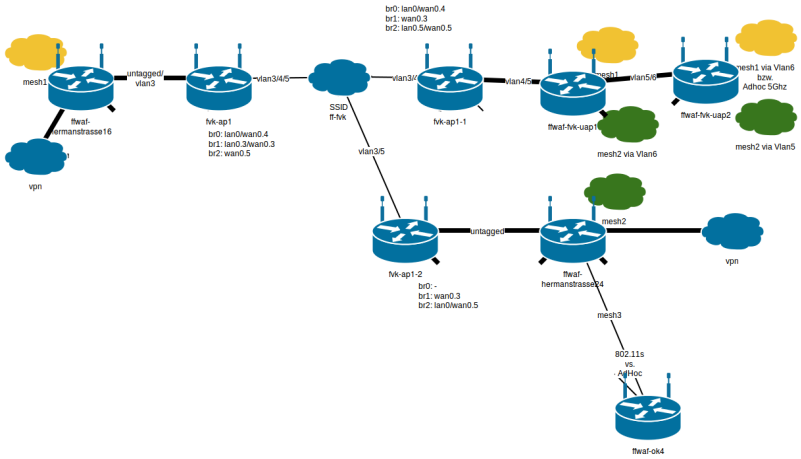
Loadsharing

Loadsharing über 2 x DSL / 2 x AP

In der Freiherr-von-Ketteler Schule Warendorf haben wir folgendes Netz gebaut:



~~In der Freifunk-Karte das aktuell hier: http://ffwaf-srv2.freifunk-muensterland.net/map/#!v:m;n:e894f6681aa4~~

An zwei Stellen wird von Nachbar mit einem TP-Link 1043 ein DSL Zugang gestellt:

- ffwaf-hermanstrasse16
- ffwaf-hermanstrasse24

In der Schule stehen zwei TP-Link 3600, die über 5GHz und inzwischen auch über LAN meshen.

- ffwaf-fvk-uap1
- ffwaf-fvk-uap2

Für die Zuführung haben wir uns für für Nano Loco entschieden, von denen je zwei in der Version M2 und zwei in der Version M5 zur Verfügung standen.

Aufgabenstellung

Aufgabenstellung war, die beiden DSL Zugänge nutzen zu können: ffwaf-hermanstrasse16 --> ffwaf-fvk-uap1 und ffwaf-hermanstrasse24 --> ffwaf-fvk-uap2

Zunächst haben wir nach reiner Lehre alle Knoten mit der aktuellen ff-Firmware geflasht. Folgende **Probleme** wurden sichtbar:

1. Wenn zwei Knoten direkt nebeneinander stehen, behindern sie sich erheblich. Ein Paket das von einem Client aufgenommen wird, wird, wenn es vom Nachbar im mesh weitergeleitet wird, im Zweifel zerstört. Im Batman äußert sich das dann mit einer sehr schlechten TQ. Abhilfe bringt den mesh in einem anderen Kanal zu transportieren. Idealerweise in einem anderen Band. Da wir 3600 mit Dual-Band im Einsatz hatten, war das zunächst die Lösung.
2. Wenn ein Knoten sowohl die Client SSID Freifunk als auch mesh anbietet, geht die Performance insgesamt in den Keller. Im Batman äußert sich das in einer sehr schlechten TQ.

Probleme

Die Probleme wurde schon sichtbar, als der DSL-Zugang an ffwaf-hermanstrasse24 noch nicht zur Verfügung stand. Um diese Probleme zu umschiffen, wurden die Nanos so konfiguriert, dass sie auf einem anderen Kanal funkten, und auch kein Freifunk anboten. Deutlich verbessert hat sich die Situation, als die bis dahin genutzten zwei M2 durch M5 Varianten (mit original Firmware) ausgetauscht wurden und so auch kein WLAN der Nachbarn mehr stören konnte. Die M5 bridgen den mesh nur. Um die M5 managen zu können, musste die Konfiguration am ffwaf-hermanstrasse16 geändert werden. Dazu später mehr.

Ein Versuch von ffwaf-hermanstrasse24 über zwei Nano M2 nach ffwaf-fvk-uap2 zu funken, hat gezeigt, dass der Link für Batman unbrauchbar wurde. Mit FF-Firmware war der TQ bei 8% (in Worten: acht) und auch mit Stock-Firmware hat ffwaf-fvk-uap2 sich in der Regel für den Weg über ffwaf-fvk-uap1 entschieden und der zweite DSL-Zugang blieb ungenutzt. Hauptursache war, dass Traffic zum LAN der Nano über einen weiteren Freifunk-Knoten (ffwaf-ok) über mesh zugeführt wurde. Bei ffwaf-ok4 handelt es sich um ein TP-Link CPE210. In diesem Setup war vier Nano im Einsatz und das Setup war weiterhin nicht gut, da die Aufgabenstellung nicht erfüllt war.

Mit der Lösung oben, für die eine dritte M5 angeschafft wurde, und ein paar weitern Kniffen in der ff-Konfiguration haben wir die Aufgabenstellung noch lösen können und zudem zwei M2 eingespart: wir legen mit VLANs und Bridges virtuelle Kabel direkt von den Knoten mit dem VPN und den Knoten mit unseren Usern.

Konfiguration ffwaf-hermanstrasse16

Um die Nanos managen zu können, muss über ffwaf-hermanstrasse16 über ein Kabel zwei Dienste anbieten: Mesh als Nutzlast und Freifunk, in dem die Nanos ihre IP-Adresse beziehen können. Die Lösung ist, diese in verschiedenen VLANs zu transportieren. Die TP-Link TL-WR1043 können tagged /untagged gleichzeitig auf einem Interface. Die Nano am Standort wird der AP.

Zunächst muss der interne Switch (port 0) tagging unterstützen. Da sich dann aber der Name der bridge ändert (von eth1 auf eth1.1) bietet sich ein reboot an:

```
uci set network.@switch_vlan[0].ports="0t 1 2 3 4"
uci set network.mesh_lan.ifname=eth1.1
uci commit
reboot
```

Jetzt wird ein neues VLAN 3 erzeugt und dem port 4 mitgeteilt, dass dort das VLAN 3 tagged anliegen soll. In dem VLAN soll das Freifunk (client) Netz eingebunden werden:

```
uci add network switch_vlan
uci set network.@switch_vlan[-1].device=switch0
uci set network.@switch_vlan[-1].vlan=3
uci set network.@switch_vlan[-1].ports="0t 4t"
uci set network.client.ifname="eth1.3 bat0"
```

Die Nano werden so konfiguriert, dass sie über VLAN3 zu managen sind. Die untagged Pakete (mit dem Mesh) werden zu einer Nano beim ffwaf-fvk-uap1 weitergeleitet.

Konfiguration ffwaf-fvk-uap1

Der ffwaf-fvk-uap1 hängt am LAN-Interface der Nano vor Ort. Da hier auch mehrere Mesh durchgeführt werden sollen, muss das Interface in den tagged-Modus versetzt werden. Im Gegensatz zu den 1043 können die TP-Link TL-WDR3600 leider entweder tagged oder untagged auf einem Interface. Auf VLAN 4 wird das mesh vom ffwaf-hermanstrasse16 aufgenommen. In VLAN 5 wird das mesh von ffwaf-hermanstrasse24 transportiert.

Vlan4

```
uci add network switch_vlan
uci set network.@switch_vlan[-1].device=switch0
uci set network.@switch_vlan[-1].vlan=4
uci set network.@switch_vlan[-1].ports="0t 2t"
uci set network.mesh_lan.ifname="eth0.4"
```

Vlan5 - eine Bridge über port2 und port3

```
uci add network switch_vlan
uci set network.@switch_vlan[-1].device=switch0
uci set network.@switch_vlan[-1].vlan=5
uci set network.@switch_vlan[-1].ports="0t 2t 3t"
```

uci set network.transp=interface

```
uci set network.transp.ifname=eth0.5
uci set network.transp.auto=1
uci set network.transp.type=bridge
uci set network.transp.macaddr=ea:95:f9:68:1a:a4
uci set network.transp.igmp_snooping=0
uci set network.transp.proto=none
```

Die MAC Adresse habe ich mir nach einem ähnlichen Schema ausgedacht, wie das im Gluon immer gemacht wird.

Das Vlan4 wird *nicht* zum ffwaf-fvk-uap2 durchgereicht. In dem Fall würde ffwaf-fvk-uap2 einen direkten Link zum ffwaf-hermanstrasse16 haben! Stattdessen wird ein neues VLAN6 erstellt und mit dem Batman verbunden:

Vlan6

```
uci add network switch_vlan
uci set network.@switch_vlan[-1].device=switch0
uci set network.@switch_vlan[-1].vlan=6
uci set network.@switch_vlan[-1].ports="0t 3t"
```

und

```
uci set network.mesh_lanz=interface
uci set network.mesh_lanz.macaddr=ea:95:f8:68:1a:a4
uci set network.mesh_lanz.mesh=bat0
uci set network.mesh_lanz.proto=batadv
uci set network.mesh_lanz.auto=1
uci set network.mesh_lanz.ifname=eth0.6
```

lanz steht für lan zwo. Mir viel auf die schnelle nichts besserer ein. Ein **batctrl if** sollte jetzt in folgendes liefern:

```
# batctl if
mesh0: active
mesh1: active
eth0.4: active
eth0.6: active
```

Also Batman auf den Radios und in Vlan4 und Vlan6.

Konfiguration ffwaf-fvk-uap2

Über Vlan5 erhalten wir den Mesh von ffwaf-hermanstrasse24:

```
uci add network switch_vlan
uci set network.@switch_vlan[-1].device=switch0
uci set network.@switch_vlan[-1].vlan=5
uci set network.@switch_vlan[-1].ports="0t 2t"
```

und (land steht für „lan drei“, weil mir wieder nix besseres eingefallen ist...)

```
uci set network.mesh_land=interface
uci set network.mesh_land.macaddr=66:67:b6:c6:f9:d2
uci set network.mesh_land.mesh=bat0
uci set network.mesh_land.proto=batadv
uci set network.mesh_land.auto=1
uci set network.mesh_land.ifname=eth0.5
```

Über Vlan6 erhalten wir den Mesh von ffwaf-hermanstrasse16:

```
uci add network switch_vlan
uci set network.@switch_vlan[-1].device=switch0
uci set network.@switch_vlan[-1].vlan=6
uci set network.@switch_vlan[-1].ports="0t 2t"
```

sowie:

```
uci set network.mesh_lanz=interface
uci set network.mesh_lanz.macaddr=66:67:b5:c6:f9:d2
uci set network.mesh_lanz.mesh=bat0
uci set network.mesh_lanz.proto=batadv
uci set network.mesh_lanz.auto=1
uci set network.mesh_lanz.ifname=eth0.6
```

Ein **batctrl if** sollte jetzt in folgendes liefern:

```
# batctl if
mesh0: active
mesh1: active
eth0.5: active
eth0.6: active
```

Um zu dem Ergebnis zu kommen, habe ich mich durch die OpenWRT Dokumentation gewühlt. Die hier wesentlichen Links sind:

- http://wiki.openwrt.org/doc/uci
- http://wiki.openwrt.org/doc/uci/network und auch http://wiki.openwrt.org/doc/uci/network/switch
- http://wiki.openwrt.org/toh/tp-link/tl-wr1043nd insb der Absatz „*WARNING: Creating a new VLAN on the 2.x router hardware (maybe 3.x too?) will move the switch into VLAN mode where the default VLAN1 needs to be referenced as interface "eth1.1", not the default "eth1". Because the default OpenWrt configuration uses the device name eth1, the result is that creating a new VLAN breaks the existing default VLAN1 "lan" interface. To avoid this, before creating a new VLAN, switch VLAN1 into tagged mode by making the CPU port tagged in VLAN1, edit the lan interface definition to be eth1.1 instead of eth1, and reboot. This is best done with SSH as both edits need to be done together and then a reboot follows. After the reboot, additional VLANs can be created as usual without disrupting VLAN1.*“
- http://wiki.openwrt.org/toh/tp-link/tl-wdr3600 insb. der Satz „*The switch driver refuses to configure a port with tagged and non-tagged VLANs. It is possible to tag multiple VLANs on the same ports*“

Die Nanos haben WDS und airMAX aktiv. Die beiden Stations können sich nicht sehen. Mit airMAX wird das "hidden node" Problem behoben: „*airMAX assigns time slots for each device communication to avoid the "hidden node" problem, which occurs when a node is visible from a wireless AP, but not from other nodes communicating with the originating AP.*“

Erstaunlicherweise hat die Konfiguration eine Migration von der Münsteraner Gluon-Version in eine Warendorfer per autoupdate überstanden.