Meshkit Documentation

Release 0.1.0

Meshkit Documentation Team

01.04.2016

KAPITEL 1

Allgemeine Infos zur Meshkit Freifunk Firmware

1.1 Was ist Meshkit?

Meshkit ist ein Generator für individualisierte Freifunk-Firmware-Images, die direkt nach dem Flashen einsatzbereit sind. Dazu gibt man im Meshkit ein paar Daten ein (IP-Adresse, Standort, Kontaktdaten). Meshkit generiert dann ein Firmwareimage (das Betriebsystem des Freifunk Routers) mit diesen Einstellungen. Dieses Image kann dann auf einen *kompatiblem Access Point* geflasht werden, um so Teil des Freifunknetzes zu werden.

1.2 Features

- Support für Communityprofile (verschiedene Default-Einstellungen für unterschiedliche Communities)
- Erstellen vorkonfigurierter Firmwareimages, die direkt nach dem Flashen im Mesh erreichbar sind.
- Pakete können direkt mit ins Image gebaut werden (das spart Platz dank besserer Kopmression)
- Eigene Dateien können mit ins Image gebaut werden

1.3 Unterstützte Routermodelle

Prinzipiell werden alle Router unterstützt, die von der von Meshkit verwendeten OpenWrt-Version unterstützt werden und für die es ein `Target` gibt. Es werden nur Router mit **mindestens 32MB Ram** unterstützt. Was die **Flash-Grösse** anbeglangt so sind 4MB gerade noch genug, empfohlen werden aber mindestens **8MB Flash**. Informationen zu den einzelnen von OpenWrt unterstützten Geräten (Target, Ram, Flash etc.) gibt es in der OpenWrt Table of Hardware.

Im Moment werden von Meshkit die folgenden Targets unterstützt:

- ar71xx

- brcm47xx
- brcm63xx
- x86
- x86-kvm

1.4 Repository

Der Quellcode von Meshkit (GPL-Lizenz) befindet sich auf Github.

Diese Dokumentation ist ebenfalls auf Github verfügbar: Meshkit Dokumentation.

Für Meshkit und die Meshkit Dokumentation gibt es ebenfalls auf Github einen Bugtracker ("Issues").

1.5 Von Meshkit benutzte Software

- Die Meshkit Weboberfläche baut auf web2py auf
- Die Firmware basiert auf einem möglichst wenig angepassten OpenWrt
- Als Weboberfläche wird LuCI verwendet
- Als Routingprotokoll kommt OLSRd zum Einsatz

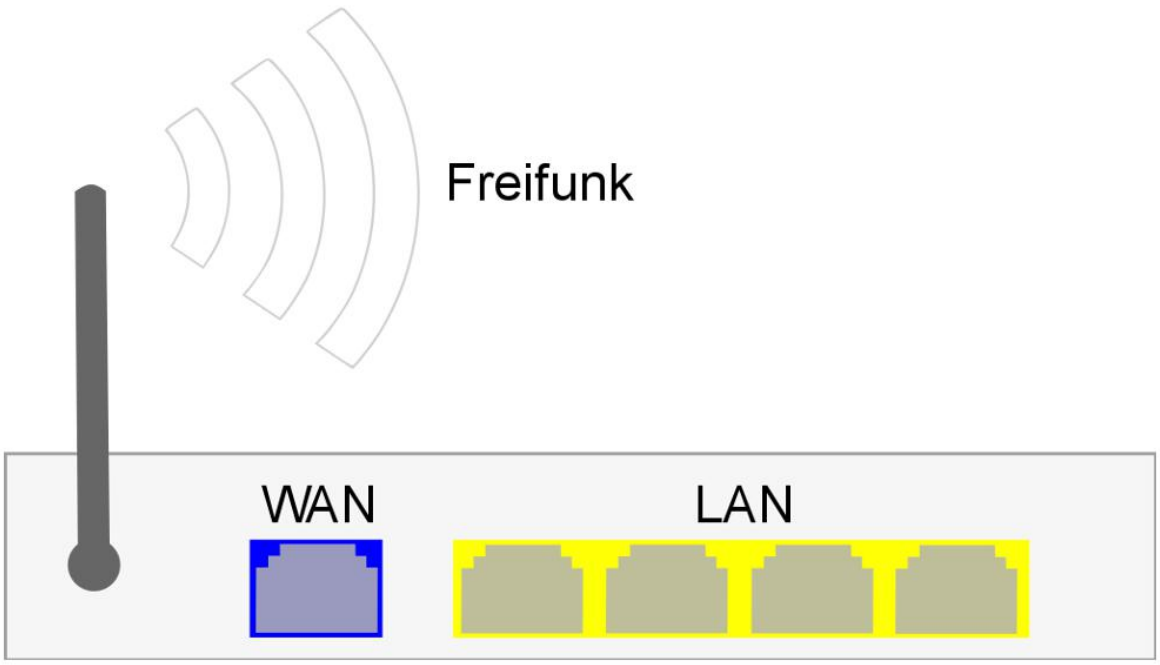
KAPITEL 2

Einbinden des Freifunkrouters ins eigene Netzwerk

In diesem Kapitel soll es draum gehen, wie ein Freifunk Router ins eigene Netzwerk integriert werden kann. Je nach bestehender Netzwerktopologie und Anforderungen kann der Freifunk Router auf unterschiedliche Art integriert werden.

Wichtig: Bevor die Meshkit Firmware generiert werden kann ist es wichtig im voraus zu wissen, wie der Freifunk Router ins eigene Netzwerk integriert werden soll.

2.1 LAN, WAN und Freifunk - Die einzelnen Zonen im Router



Bevor einzelne Integrationsvarianten besprochen werden ist es wichtig zu verstehen, dass der Freifunk Router in der Standardkonfiguration drei Netzwerke kennt. Jedes dieser drei Netzwerke erfüllt unterschiedliche Aufgaben und es gelten unterschiedliche Regeln, vor allem im

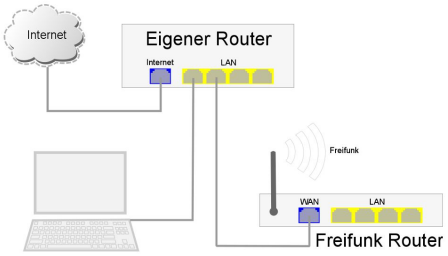
Hinblick auf die Konfiguration der Firewall.

Nicht jeder Router verfügt über genug Schnittstellen um alle drei Zonen zur Verfügung zu stellen. Gibt es z.B. nur eine Ethernet-Buchse am Router dann wird durch OpenWrt vorgegeben, welchem Netzwerk diese Buchse angehört. In der Regel konfiguriert OpenWrt diese jedoch dann als LAN. Genaueres dazu findest du in der Table of Hardware im OpenWrt Wiki.

Zone	Aufgabe	Default IP
LAN	Dies ist das lokale Netzwerk (Local Area Network). Hier werden eigene PCs angeschlossen.	192.168.1.1
WAN	Wide Area Network. Damit verbindet man den Router mit dem Internet.	automatisch (*DHCP*)
Freifunk	Für die Verbindung zum Freifunknetz	individuell

Für die einzelnen Zonen gelten spezifische **Firewallregeln**, die bestimmen, ob Verkehr der den Router über diese Zonen erreicht akzeptiert (und evtl. weitergeleitet) ode verworfen wird, siehe dazu: *Firewallzonen*.

2.2 Der Standardfall: Freifunk Router am Heimnetzwerk



Der Freifunkrouter ist mit seiner WAN-Buchse am LAN des lokalen Router/Switch angeschlossen und damit Teil des eigenen Netzwerks. Eigene PCs sind ebenfalls mit diesem Router/Switch verbunden.

- PC und Freifunk Router sind im selben Netzwerk (LAN)
- Beide beziehen in der Regel automatisch eine IP-Adresse vom eigenen Router (*DHCP*)
- Der eigene Router dient als Internetgateway
- Ob auch Freifunknutzer den eigenen Internetzugang benutzen dürfen kann im Meshkit mit der Option `Internet freigeben` (siehe *Konfiguration von WAN im Meshkit*) konfiguriert werden.
- Auf Rechner im LAN kann von Freifunk aus nicht direkt zugegriffen werden. Will man Dienste auf Rechnern im LAN verfügbar machen muss man dies auf dem Freifunkrouter konfigurieren, siehe *Angeschlossene Computer im Freifunknetz erreichbar machen*.
- Um vom WAN aus auf den Freifunk Router zugreifen zu können, muss die Firewall dort konfiguriert werden. Das kann man sich einfach machen indem man beim meshkit im `WAN` Tab Häckchen bei `Erlaube SSH` und `Erlaube Web` setzt (siehe *Konfiguration von WAN im Meshkit*)

- Will man vom eigenen LAN aus auf das Freifunknetz zugreifen können, dann müssen entweder auf dem eigenen Rechner oder besser auf dem eigenen Gateway statische Routen eingetragen und die Firewall auf dem Freifunk Router entsprechend konfiguriert werden (siehe *Routing von WAN nach Freifunk ermöglichen*) werden.

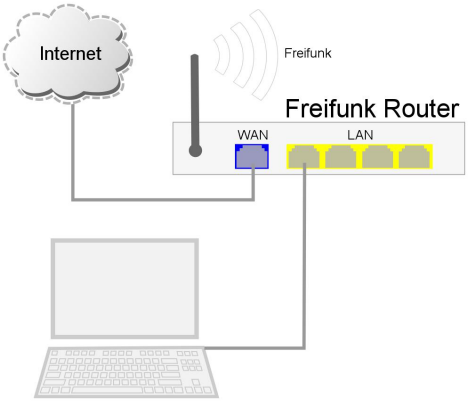
2.2.1 Routing von WAN nach Freifunk ermöglichen

Folgende statische Routen sollten auf dem eigenen Router (oder notfalls auf dem eigenen Rechner) konfiguriert werden, das Handbuch des eigenen Routers hilft hier sicher weiter:

Netzwerk	Netzmaske	Beschreibung
10.0.0.0	255.0.0.0 (/8)	Freifunk Netze
172.22.0.0	255.254.0.0 (/15)	DN42
172.31.0.0	255.255.0.0 (/16)	ChaosVPN

Per Default darf Traffic, der an der WAN-Schnittstelle ankommt nicht ins Freifunknetz geroutet werden. Damit das dennoch möglich ist, ändert man am einfachsten die **Einstellungen für die Firewallzone** `wan` und setzt dort alles (INPUT, FORWARD, OUTPUT) auf *annehmen* (ACCEPT). Siehe *Firewallzonen*.

2.3 Freifunk Router direkt am Internetzugang



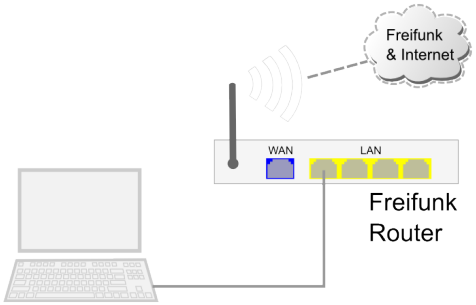
Der Router ist mit seinem WAN-Port direkt am Internet (z.B. einem DSL-Modem) angeschlossen und stellt die Verbindung zum Internet selbst her. Eigene PCs werden an den LAN-Ports des Routers angeschlossen oder verbinden sich per WLAN. Auf diese Weise kann der Freifunk Router unter Umständen auch ein bereits vorhandenes Gerät ersetzen.

- Auf Rechner im LAN kann von Freifunk aus nicht direkt zugegriffen werden. Will man Dienste auf Rechnern im LAN verfügbar machen muss man dies auf dem Freifunkrouter konfigurieren, siehe *Angeschlossene Computer im Freifunknetz erreichbar machen*.
- Rechner die am LAN angeschlossen sind können andere Adressen im Freifunknetzwerk direkt erreichen
- Wenn die Hardware des Freifunkrouters das Erstellen von Virtuellen Access Points (VAP) erlaubt, dann kann zusätzlich zum Freifunknetz ein weiteres verschlüsseltes

WLAN-Netz für die eigene Benutzung geöffnet werden. Siehe *Privates WLAN-Netzwerk einrichten*.

Hinweis: Viele Provider benutzen VOIP (SIP) zum Bereitstellen von Telefoniefunktionen. Dabei kümmert sich der vom Provider zur Verfügung gestellte Router um die VOIP- Verbindungen. Ist dies bei dir der Fall, dann ist dieses Setup eher nichts für dich und es ist wahrscheinlich sinnvoller, nach *Der Standardfall: Freifunk Router am Heimnetzwerk* vorzugehen.

2.4 Freifunk Router ohne eigenen Internetzugang



Ein Freifunkrouter lässt sich auch ohne Anschluss an einen Internetzugang bzw. ans eigene Heimnetz betreiben. Eigene Rechner sind in diesem Fall an den LAN-Schnittstellen des Freifunkrouters angeschlossen. Voraussetzung ist, dass über Funk eine Verbindung zum restlichen Teil des Freifunknetzwerks möglich ist, sich also mindestens ein Nachbar in Funkreichweite befindet. Ist dies der Fall, dann wird über diese Funkverbindung Kontakt zum Freifunknetzwerk hergestellt. Damit ist dann in der Regel auch der Zugriff aufs Internet vom Router selbst und angeschlossenen Rechnern möglich.

- Auf Rechner im LAN kann von Freifunk aus nicht direkt zugegriffen werden. Will man Dienste auf Rechnern im LAN verfügbar machen muss man dies auf dem Freifunkrouter konfigurieren, siehe *Angeschlossene Computer im Freifunknetz erreichbar machen*
- Rechner die am LAN angeschlossen sind können andere Adressen im Freifunknetzwerk direkt erreichen

KAPITEL 3

Firmwareimage mit Meshkit generieren

In diesem Kapitel wird erklärt, wie man mit Meshkit eine komplette und fertig eingerichtete Firmware für den eigenen Router generiert.

Inhalt:

3.1 Bevors los geht: Vorbereitungen

3.1.1 Wichtige Infos zum Router sammeln

Um eine passende Firmware für das eigene Routermodell zu generieren muss man zunächst einige Daten zum Router kennen:

- Chipsatz
- Chipsatz des/der WLAN-Interfaces
- Größe des RAM
- Größe des Flash-Speichers

Diese Daten finden man in der Table of Hardware im OpenWrt Wiki. Zu den meisten Routermodellen gibt es auch eine Detailseite mit weiteren Infos. Diese Seite sollte man ebenfalls zumindest kurz überfliegen. Hier finden sich im Allgemeinen auch Infos zum Flashen und zur Rettung des Routers falls beim Flashen etwas so richtig schief ging.

Beispiel:

Wir wollen die Daten für einen TP-Link WR1043ND zusammentragen. In der Table of Hardware sehen wir folgendes:

Model	Version	Status	Target(s)	Platform	CPU Speed (MHz)	Flash (MB)	RAM (MB)	Wireless NIC	Wireless Standard	Detachable Antennae	Wir Por
TL-WR1043ND	1-1.8	10.03.1	ar71xx	Atheros AR9132	400	8	32	Atheros AR9100 (integrated)	11b/g/n	Yes x3	5 gig
TL-WR2543ND	1.0	12.09	ar71xx	Atheros AR7242	400	8	64	Atheros AR9380 (onboard)	11a/b/g/n	Yes x3	5 gig

Daraus können wir entnehmen:

- Der Chipsatz gehört zur AR71XX Familie
- WLAN benutzt Atheros Hardware
- Der RAM ist 32 MB groß
- Flash ist 8 MB groß

3.1.2 IP-Adresse(n) registrieren

Nun muss noch mindestens eine IP-Adresse aus dem Freifunknetz für den Router registriert werden. Hier hat jede Community ihre eigenen Seiten zur Registrierung, im Allgemeinen kann man sich hier jedoch selbst bedienen und einfach eine noch nicht vergebene Adresse für seinen Knoten reservieren.

Registrierungsseiten in den einzelnen Communities:

- **Augsburg:** Als Benutzer auf http://augsburg.freifunk.net einloggen und "Node registrieren"

3.2 Für eilige: Quickstart

Dies ist nur eine kurze Anleitung wie man am schnellsten ein Firmwareimage mit Meshkit generiert. Für eine ausführliche Anleitung und Erklärung aller Optionen im Detail siehe: *Firmware mit Meshkit generieren lassen*.

3.2.1 1. Grundlegende Einstellungen

Gehe zur Meshkit-Seite. Wähle dort eine Community und ein Target. Target kann durch Klick auf eines der Modelle unten auch automatisch gewählt werden. Willst du eine Email nach dem Bauen der Firmware bekommen dann gib noch eine Email-Adresse an.

3.2.2 2. Konfiguration der Firmware

Wähle ein Profil. Hast du im vorigen Schritt bereits ein Modell aus der Liste unten gewählt ist hier schon das richtige ausgewählt.

Fülle die Felder in den Tabs `Standort` und `Kontakt` aus.

Gib im Tab `Wireless` eine IP-Adresse aus dem Netzwerk deiner Community ein. Siehe: *IP-Adresse(n) registrieren*.

Jetzt noch das Formular absenden damit die Firmware generiert wird.

3.2.3 3. Firmware erstellen

Das dauert jetzt etwa 20 Sekunden. Wenn die Firmware erstellt wurde werden ein paar Links angezeigt. Wähle das für deinen Router passende Image (siehe Table of Hardware) und flashe es auf den Router. Siehe *Meshkit Firmware auf einen Router flashen*

3.3 Firmware mit Meshkit generieren lassen

Das Erstellen von Firmwareimages erfolgt in drei Schritten:

3.3.1 1. Grundlegende Systemeinstellungen

Öffne die Meshkit Webseite im Browser. Du siehst nun folgendes:

Erklärung der einzelnen Optionen

`1. Login bzw. Registrieren`

Hier kann man einen User für Meshkit registrieren. Dadurch wird es möglich, einige Datenfelder beim Generieren neuer Images bereits auszufüllen, z.B. Community, Adresse oder SSH Public Keys. Es ist nicht notwendig einen Benutzer im Meshkit zu registrieren. Wer aber öfter Images generiert dem kann die Registrierung ersparen, bei einigen Feldern immer und immer wieder die selben Daten einzugeben.

`2. Geräte Vorauswahl`

Im unteren Bereich von Meshkit befinden sich einige Links zu Geräten, die häufig verwendet

werden. Klickt man auf einen dieser links, dann wählt Meshkit automatisch das richtige Target (und in Schritt 2 auch das passende Profil) für dieses Gerät. Wenn ein Gerät nicht in dieser Liste steht dann heisst das nicht, dass es nicht unterstützt wird, sondern nur, dass man Target und im nächsten Schritt Profil manuell auswählen muss.

`3. Einstellungen`

Option	Beschreibung
Community	Wähle deine Community aus. Wenn es noch keine Community für deine Stadt gibt dann siehe *Profil für eine neue Community erstellen*.
Target	Wähle ein passendes Target (siehe *Wichtige Infos zum Router sammeln*)
Expertenmodus	Wenn ausgewählt, dann werden im nächsten Schritt wesentlich mehr Optionen zur Konfiguration des Routers angezeigt.
Keine Konfiguration	Erstellt ein Image, es wird aber keine Konfiguration durchgeführt. Dies ist vor allem nützlich um Images zu erhalten, die für sysupgrade (TODO: Seite zu sysupgrade anlegen) verwendet werden können.
Email	Wenn angegeben wird nachdem das Image gebaut wurde eine Mail an diese Adresse geschickt.

`4. Absenden`

Nachdem alle Einstellungen getätigt wurde klicke auf **Absenden** um zu Schritt 2 des Meshkits zu gelangen, wo weitere Einstellungen vorgenommen werden müssen.

3.3.2 2a Konfiguration der Firmware (Normalmodus)

In diesem Schritt geht es an die eigentliche Konfiguration der Firmware im `Normalmodus`. Wenn du im vorherigen Kapitel die Option `Èxperteneinstellungen` gewählt hast dann gehe gleich weiter zum nächsten Kapitel.

Auf der Seite sind nun einige Tabs zu sehen. Diese kann man nacheinander durchgehen und alle Einstellungen vornehmen.

Hinweis: Viele Felder sind schon mit sinnvollen Optionen vorbelegt. Ist man als User im Meshkit eingeloggt werden auch Kontaktdaten etc. vorausgefüllt.

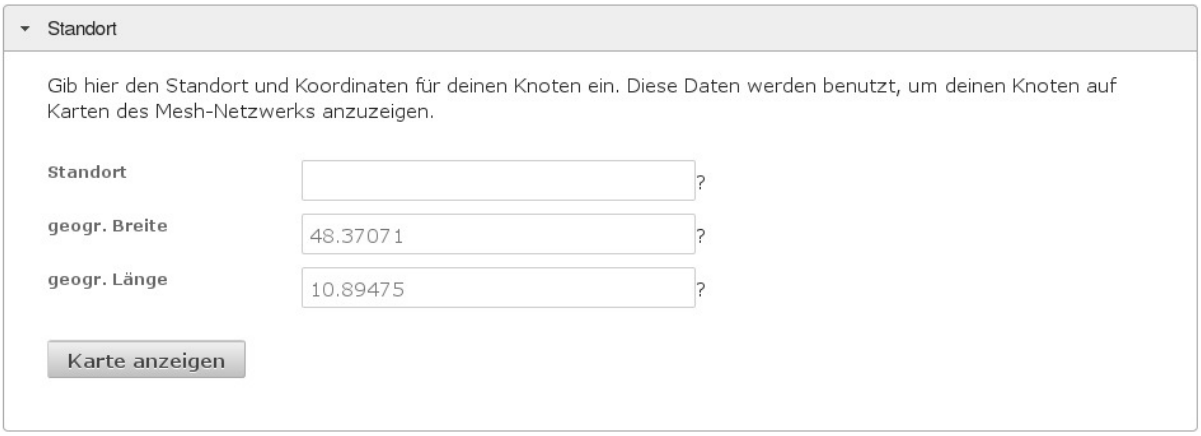
Systemeinstellungen

Hier können allgemeine Systemeinstellungen vorgenommen werden.

Option	Beschreibung	Default
Profil	Wählt ein Hardwareprofil für den Router aus. Wenn Du im vorigen Schritt ein Routermodell aus der Liste unten ausgewählt hast, dann ist das Profil bereits richtig ausgewählt. Ansonsten wähle hier das zu deinem Router passende Profil. Siehe dazu die OpenWrt Table of Hardware	generisches Profil
Name des Knotens	Ein im Netzwerk eindeutiger Name für den Knoten.	IP-Adresse des ersten Wifi Interfaces. Punkte werden dabei durch Striche ersetzt, z.B. 10-0-0-1

Standort

Hier sollten ein paar Daten zum Standort des Routers angegeben werden. Länge und Breite sind notwendig, um den Router automatisch auf Karten des Netzwerks platzieren zu können.

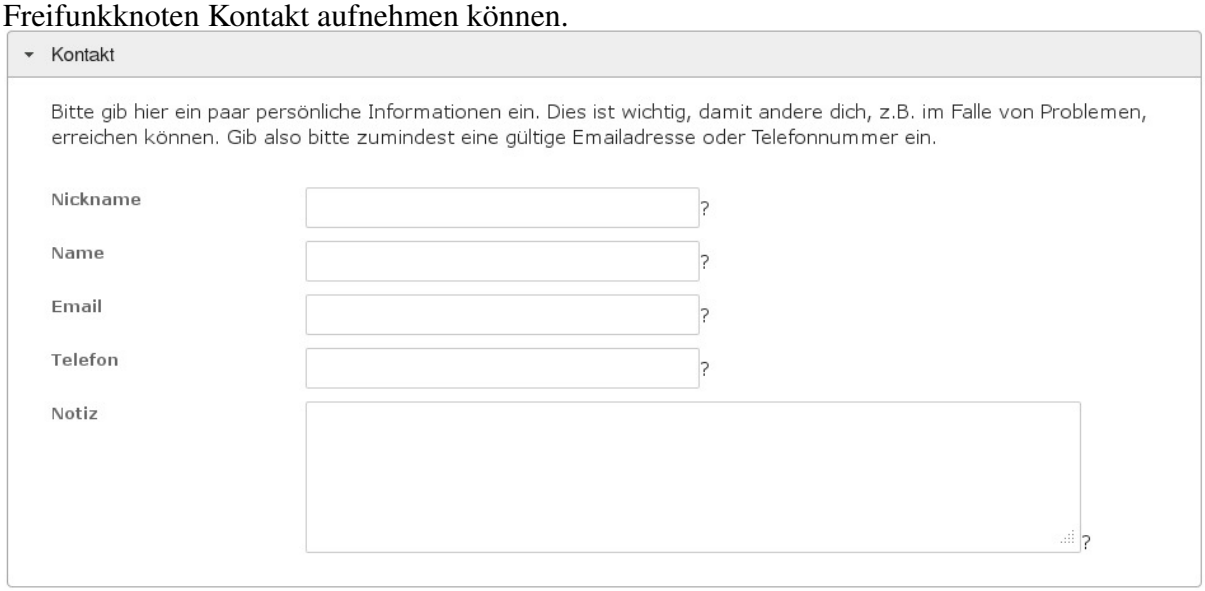


Klickt man auf den `Karte anzeigen` Button öffnet sich eine Karte, in der man durch klicken auf einen Punkt die Werte `geogr. Länge` und `geogr. Breite` automatisch setzen kann.

Option	Beschreibung	Default
Standort	Standort des Routers als Text, z.B. Strasse und Ort	(keiner)
geogr. Länge	Längenangabe zur Position des Routers	definiert durch Community-Profil, siehe *Community Profile*
geogr. Breite	Breitenangabe zur Position des Routers	definiert durch Community-Profil, siehe *Community Profile*

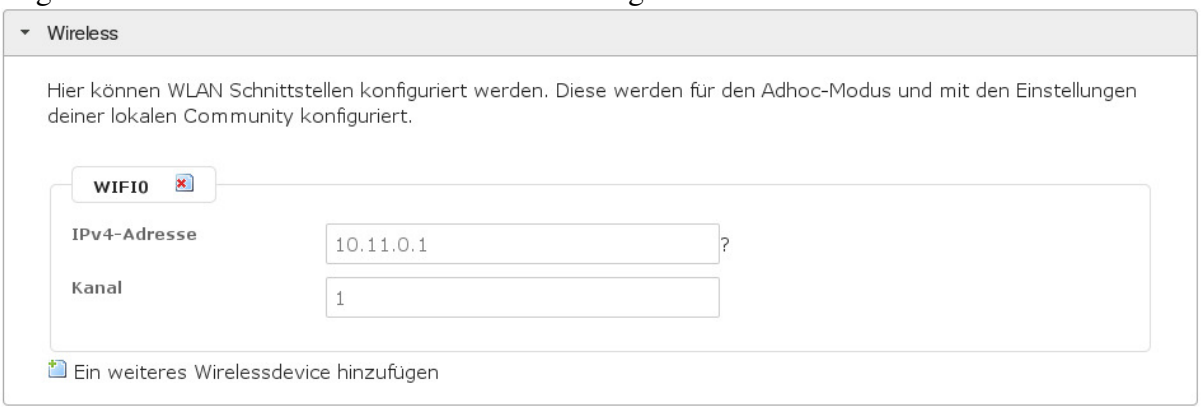
Kontakt

Hier werden Kontakteinstellungen usw. vorgenommen, die auch auf der öffentlichen Webseite des Freifunk Routers angezeigt werden. Es ist empfehlenswert zumindest einen Nickname und eine gültige Emailadresse anzugeben, damit interessierte Nutzer oder andere Betreiber von Freifunkknoten Kontakt aufnehmen können.



Wireless

Einstellungen für ein oder mehrere WLAN-Schnittstellen. Für Geräte mit mehreren WLAN-Interfaces können durch klicken auf `Ein weiteres Wirelessdevice hinzufügen` insgesamt bis zu drei WLAN-Schnittstellen konfiguriert werden.



Option	Beschreibung	Default
IP-Adresse	eine netzweit eindeutige IP-Adresse die in den meisten Communities zentraler Stelle vergeben wird. Siehe: *IP-Adresse(n) registrieren*. **Die IP sollte auf jeden Fall geändert werden.**	Erste IP im Netzwerk der Community.
Kanal	Der Funkkanal. Hier steht per Default der richtige Kanal für die gewählte Community. Ändere den Kanal daher nur wenn du genau weisst was du tust.	Voreinstellung aus Community Profil, siehe *Community Profile*

Abschicken

Nachdem alle Einstellungen getätigt sind drücke im Tab `Abschicken` noch auf `Submit`. Sind alle Eingaben gültig kommst du zum letzten Schritt des Meshkit, in dem deine Firmware gebaut wird. Bei ungültigen Eingaben weisst Meshkit darauf hin, welche Felder ungültig sind.

-> Weiter zu *3. Firmware erstellen*

3.3.3 2b Konfiguration der Firmware (Expertenmodus)

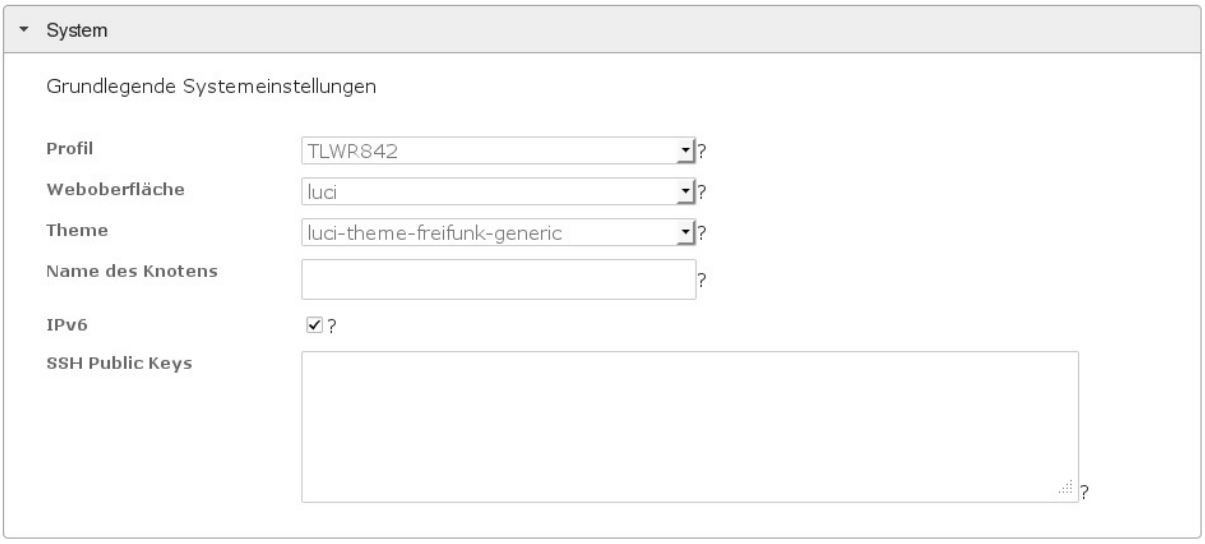
In diesem Schritt geht es an die eigentliche Konfiguration der Firmware im `Expertenmodus`. Es ist möglich, hier mehr Einstellungen selbst anzupassen.

Auf der Seite sind nun einige Tabs zu sehen. Diese kann man nacheinander durchgehen und alle Einstellungen vornehmen.

Hinweis: Viele Felder sind schon mit sinnvollen Optionen vorbelegt. Ist man als User im Meshkit eingeloggt werden auch Kontaktdaten etc. vorausgefüllt.

Systemeinstellungen

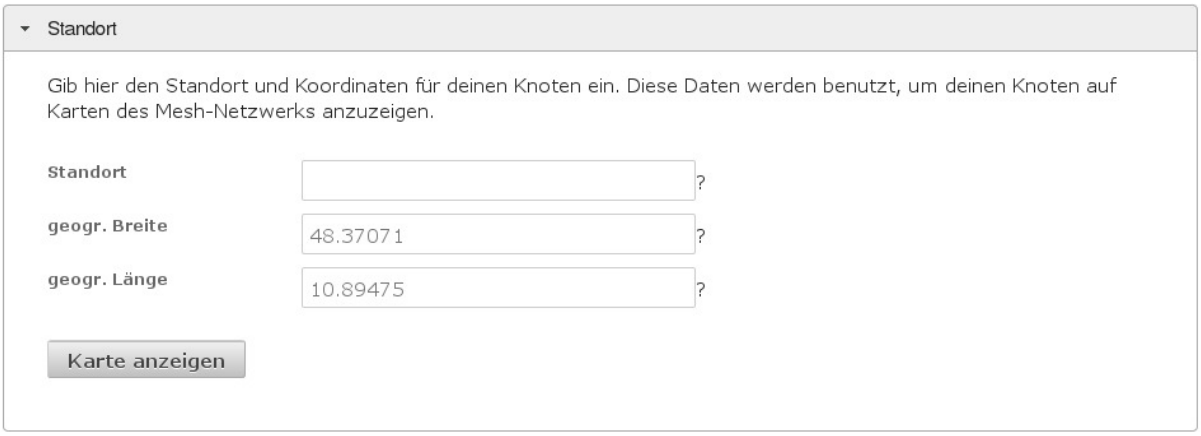
Hier können allgemeine Systemeinstellungen vorgenommen werden.



Option	Beschreibung	Default
Profil	Wählt ein Hardwareprofil für den Router aus. Wenn Du im vorigen Schritt ein Routermodell aus der Liste unten ausgewählt hast, dann ist das Profil bereits richtig ausgewählt. Ansonsten wähle hier das zu deinem Router passende Profil. Siehe dazu die OpenWrt Table of Hardware	generisches Profil
Weboberfläche	Die zu installierende Weboberfläche. Wird 'none' gewählt dann wird keine Weboberfläche gewählt	LuCI
Theme	Zu installierendes Theme. Nur Freifunk-Generic hat einige Anpassungen damit die Oberfläche für verschiedene Communities angepasst werden kann und ist daher das empfohlene Theme.	freifunk-generic
Name des Knotens	Ein im Netzwerk eindeutiger Name für den Knoten.	IP-Adresse des ersten Wi-fi Interfaces. Punkte werden dabei durch Striche ersetzt, z.B. 10-0-0-1
IPv6	IPv6 aktivieren. Wird nur angezeig wenn IPv6 im Communityprofil aktiviert ist.	durch Communityprofil vorgegeben, siehe *Community Profile*
SSH Public Keys	SSH Public Keys, einer pro Zeile. Rechner, deren öffentlicher SSH-Schlüssel hier gepeichert ist können sich mit diesem Key beim SSH-Server authentifizieren	keine

Standort

Hier sollten ein paar Daten zum Standort des Routers angegeben werden. Länge und Breite sind notwendig, um den Router automatisch auf Karten des Netzwerks platzieren zu können.

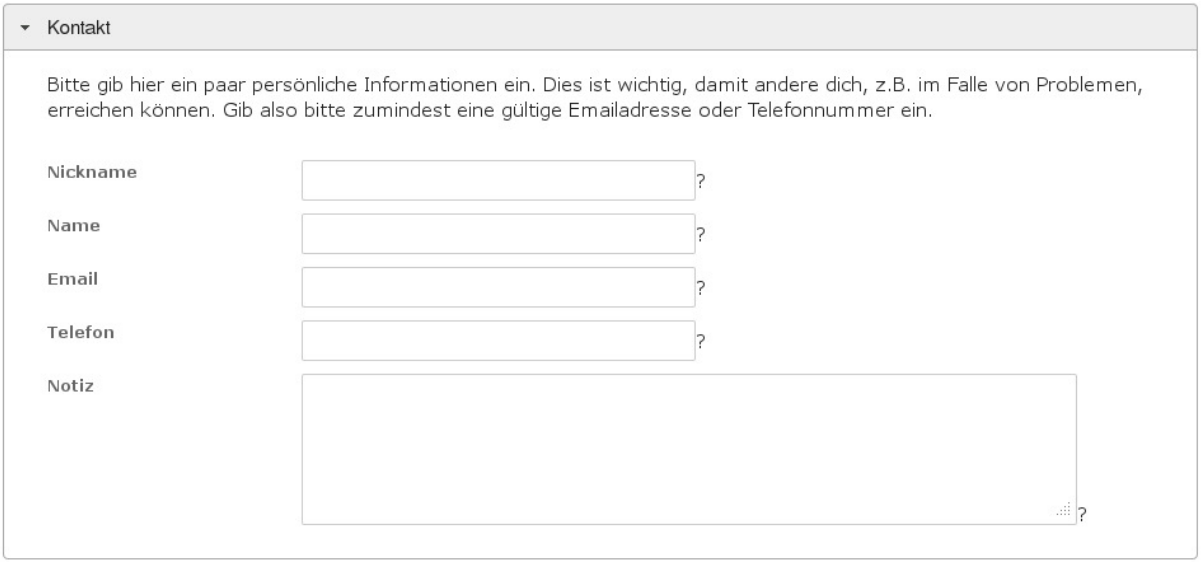


Klickt man auf den `Karte anzeigen` Button öffnet sich eine Karte, in der man durch klicken auf einen Punkt die Werte `geogr. Länge` und `geogr. Breite` automatisch setzen kann.

Option	Beschreibung	Default
Standort	Standort des Routers als Text, z.B. Strasse und Ort	(keiner)
geogr. Länge	Längenangabe zur Position des Routers	definiert durch Community-Profil, siehe *Community Profile*
geogr. Breite	Breitenangabe zur Position des Routers	definiert durch Community-Profil, siehe *Community Profile*

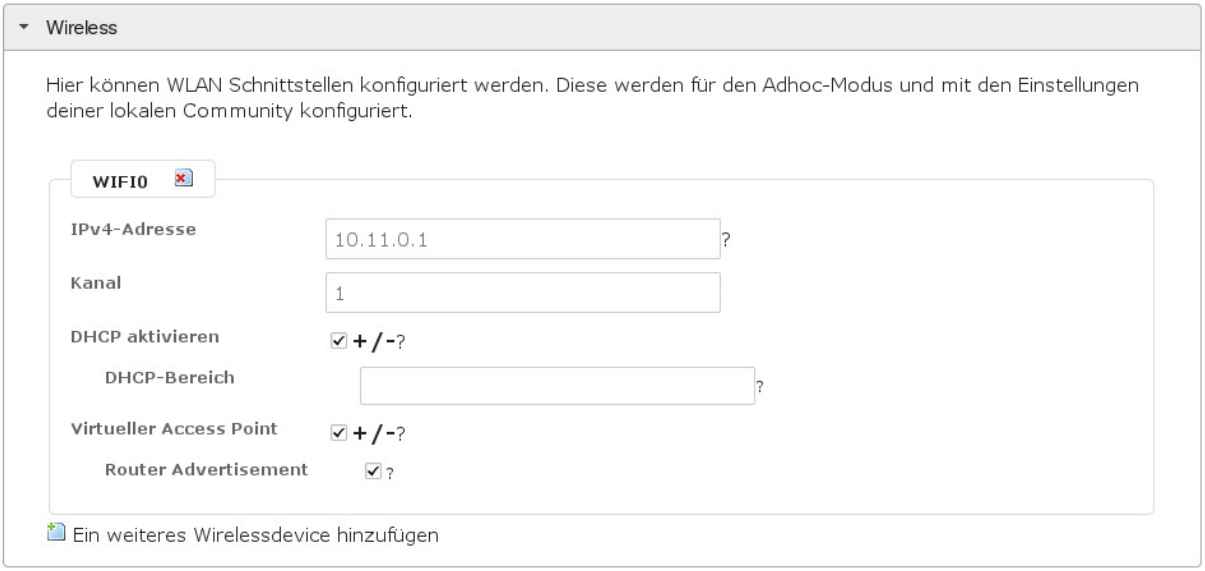
Kontakt

Hier werden Kontakteinstellungen usw. vorgenommen, die auch auf der öffentlichen Webseite des Freifunk Routers angezeigt werden. Es ist empfehlenswert zumindest einen Nickname und eine gültige Emailadresse anzugeben, damit interessierte Nutzer oder andere Betreiber von Freifunkknoten Kontakt aufnehmen können.



Wireless

Einstellungen für ein oder mehrere WLAN-Schnittstellen. Für Geräte mit mehreren WLAN-Interfaces können durch klicken auf `Ein weiteres Wirelessdevice hinzufügen` insgesamt bis zu drei WLAN-Schnittstellen konfiguriert werden.

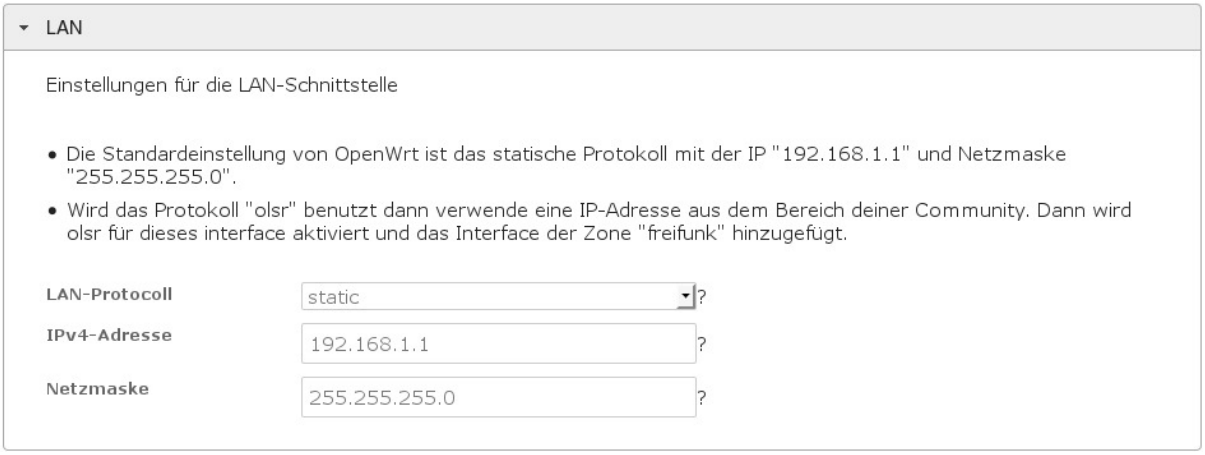


Option	Beschreibung	Default
IP-Adresse	eine netzweit eindeutige IP-Adresse die in den meisten Communities zentraler Stelle vergeben wird. Siehe: *IP-Adresse(n) registrieren*. **Die IP sollte auf jeden Fall geändert werden.**	Erste IP im Netzwerk der Community.
Kanal	Der Funkkanal. Hier steht per Default der richtige Kanal für die gewählte Community. Ändere den Kanal daher nur wenn du genau weisst was du tust.	Voreinstellung aus Community Profil, siehe *Community Profile*
DHCP aktivieren	Per *DHCP* kann Gästen im WLAN automatisch eine IP Adresse zugewiesen werden.	ja
DHCP-Bereich	Der IP-Bereich aus dem Adressen für *DHCP* vergeben werden. Liegt dieser Bereich innerhalb des im Communityprofil angegebenen Meshnetzwerks (siehe *Community Profile*), dann wird er als *HNA* von olsrd angekündigt. Liegt er ausserhalb, dann wird NAT aktiviert. Wird dieses Feld leer gelassen wird automatisch ein IP-Bereich aus 6.0.0.0/8 generiert.	Netzwerk aus 6.0.0.0/8. Dieses Netzwerk (/24) wird automatisch aus der IPv4-Adresse dieses Interfaces generiert.
Virtueller Access Point	Zusätzlich zum Mesh-Interface im adhoc-Modus wird ein weiteres WLAN-Interface im Access Point-Modus erstellt. Dies ist nützlich, damit sich z.B. auch Clients, die kein adhoc beherrschen (z.B. Android-Geräte) mit dem Netzwerk verbinden können. **Dies funktioniert nur mit Treibern die das unterstützen. Im Moment sind das madwifi, ath5k und ath9k.**	Voreinstellung aus Community Profil, siehe *Community Profile*
Router Advertisements	Wenn aktiviert dann werden auf dem Virtuellen Access Point Router Advertisements zur automatischen Konfiguration von IPv6 beim Client verschickt.	Voreinstellung aus Community Profil, siehe *Community Profile*

Konfiguration von LAN im Meshkit

Hier kann die LAN-Schnittstelle des Routers konfiguriert werden.

Hinweis: Wenn der Router mit seiner WAN-Schnittstelle an einem Netzwerk angeschlossen wird das den IP-Bereich 192.168.1.0/24 benutzt, dann **muss** hier eine IP aus einem anderen Subnetz (z.B. 192.168.2.0/24) vergeben werden.

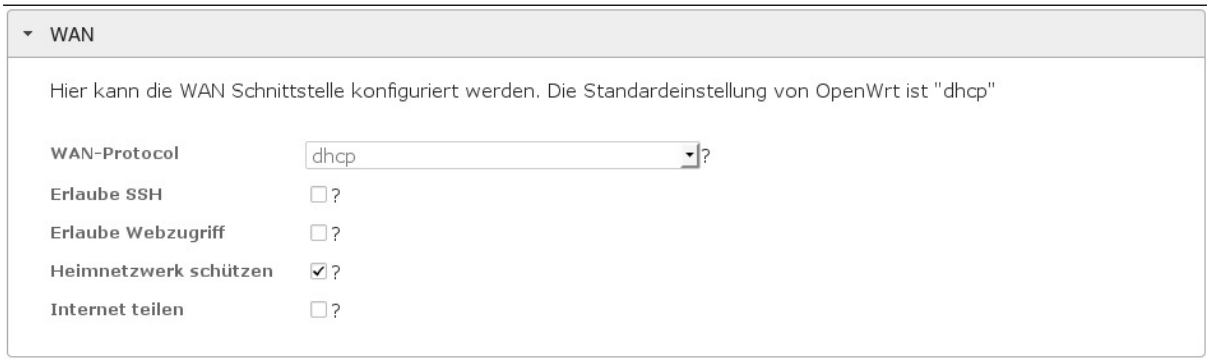


Option	Beschreibung	Default
LAN-Protokoll	Hier kann gewählt werden, ob die LAN Schnittstelle als Schnittstelle für *OLSR* konfiguriert werden soll. Wird `olsr` als Protokoll gewählt dann können an diese Schnittstelle weitere Freifunk Router per Kabel angeschlossen werden.	static
IPv4-Adresse	Die IPv4-Adresse der LAN-Schnitstelle. Wurde als Protokoll `olsr` gewählt dann sollte hier eine IP aus dem Mesh-Netzwerk der Community gewählt werden. Siehe: *IP-Adresse(n) registrieren*	192.168.1.1
Netzmaske	Netzmaske für das LAN-Interface. Soll dieses Netzwerk ein *OLSR*-Netzwerk sein, dann hat es sich als praktisch erwiesen, hier kleinere Netzmasken für die lokale Vernetzung von Nodes zu verwenden.	255.255.255.0
DHCP aktivieren	Nur verfügbar wenn `olsr` als Protokoll gewählt wurde. Per *DHCP* kann Gästen im Netzwerk automatisch eine IP Adresse zugewiesen werden.	nein
DHCP-Bereich	Der IP-Bereich aus dem Adressen für *DHCP* vergeben werden. Liegt dieser Bereich innerhalb des im Communityprofil angegebenen Meshnetzwerks (siehe *Community Profile*), dann wird er als *HNA* von olsrd angekündigt. Liegt er ausserhalb, dann wird NAT aktiviert. Wird dieses Feld leer gelassen wird automatisch ein IP-Bereich aus 6.0.0.0/8 generiert.	Netzwerk aus 6.0.0.0/8. Dieses Netzwerk (/24) wird automatisch aus der IPv4-Adresse dieses Interfaces generiert.

Konfiguration von WAN im Meshkit

Hier kann die LAN-Schnittstelle des Routers konfiguriert werden.

Hinweis: Wenn der Router mit seiner WAN-Schnittstelle an das eigene Netzwerk angeschlossen werden soll (siehe: *Der Standardfall: Freifunk Router am Heimnetzwerk*), dann solltest du hier den Zugriff auf die Weboberfläche und SSH erlauben, damit du aus deinem Netzwerk auf den Router zugreifen kannst.



Es stehen insgesamt drei Protokolle zur Auswahl:

- dhcp - Automatische Konfiguration der WAN-Schnittstelle durch *DHCP*
- static - Statische konfiguration
- olsr - Als OLSR-Schnittstelle konfigurieren (z.B. für Kabelkopplung von Nodes)

Optionen für Protokoll dhcp

Option	Beschreibung	Default
Erlaube SSH	Zugriff auf SSH (Port 22) in der Firewall erlauben	nein
Erlaube Web	Zugriff auf die Weboberfläche (Ports 80 und 443) in der Firewall erlauben	nein
Heimnetzwerk schützen	Verhindert Zugriffe aus dem Mesh auf das lokale Netzwerk hinter der WAN-Schnittstelle	ja
Internet teilen	Anderen im Mesh erlauben die eigene Internetverbindung mitzunutzen. Ist diese Option aktiviert dann wird das olsrd dyngw_plain Plugin aktiviert. Dieses überwacht die Internetverbindung und kündigt einen verfügbaren Internetzugang als *HNA* an sobald er erkannt wurde. **Wenn die eigene Internetverbindung nur über einen Tunnel freigegeben werden soll, dann diese Option nicht wählen.**	nein

Optionen für Protokoll static

Option	Beschreibung	Default
IPv4 Adresse	IPv4 Adresse der WAN-Schnittstelle	keine
Netzmaske	Netzmaske der WAN-Schnittstelle, z.B. 255.255.255.0	keine
Gateway	Gateway des Netzwerks	keines
DNS	DNS Server. Mehrere Server durch Leerzeichen voneinander getrennt angeben.	keines
Erlaube SSH	Zugriff auf SSH (Port 22) in der Firewall erlauben	nein
Erlaube Web	Zugriff auf die Weboberfläche (Ports 80 und 443) in der Firewall erlauben	nein
Heimnetzwerk schützen	Verhindert Zugriffe aus dem Mesh auf das lokale Netzwerk hinter der WAN-Schnittstelle	ja
Internet teilen	Anderen im Mesh erlauben die eigene Internetverbindung mitzunutzen. Ist diese Option aktiviert dann wird das olsrd dyngw_plain Plugin aktiviert. Dieses überwacht die Internetverbindung und kündigt einen verfügbaren Internetzugang als *HNA* an sobald er erkannt wurde. **Wenn die eigene Internetverbindung nur über einen Tunnel freigegeben werden soll, dann diese Option nicht wählen.**	nein

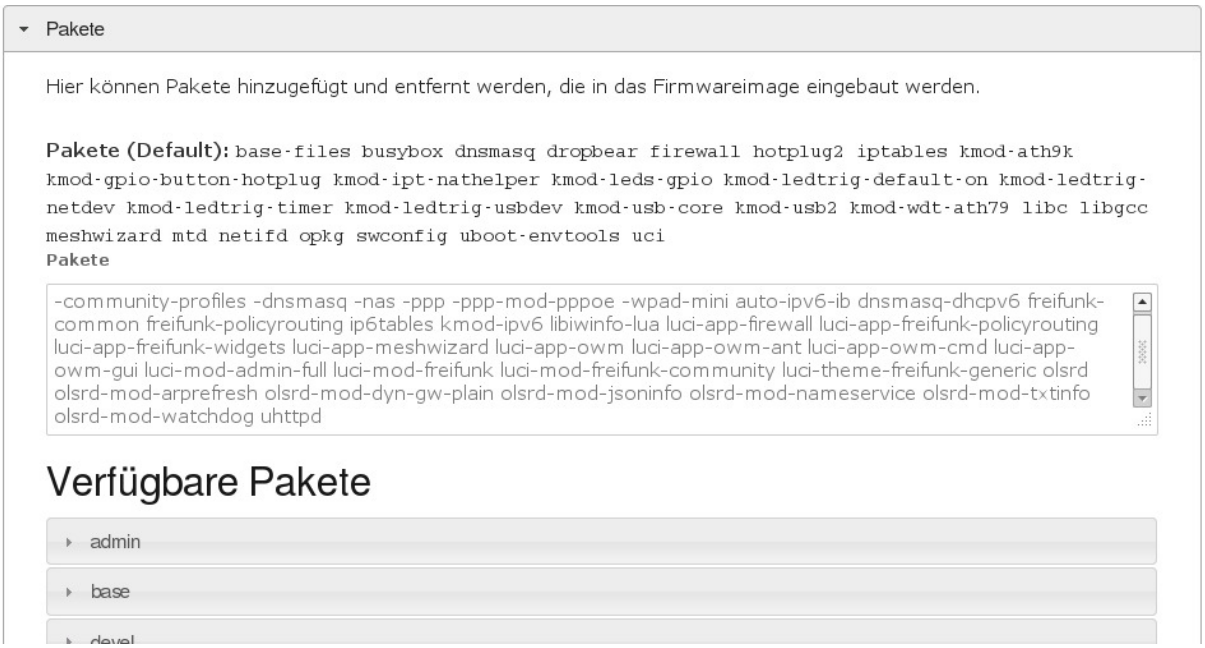
Optionen für Protokoll olsr

Option	Beschreibung	Default
IPv4-Adresse	Eine IP aus dem Mesh-Netzwerk der Community. Siehe: *IP-Adresse(n) registrieren*	keine
Netzmaske	Netzmaske für das WAN-Interface. Es sich als praktisch erwiesen, für kabelkopplungen kleinere Netzmasken zu verwenden.	keine

Pakete

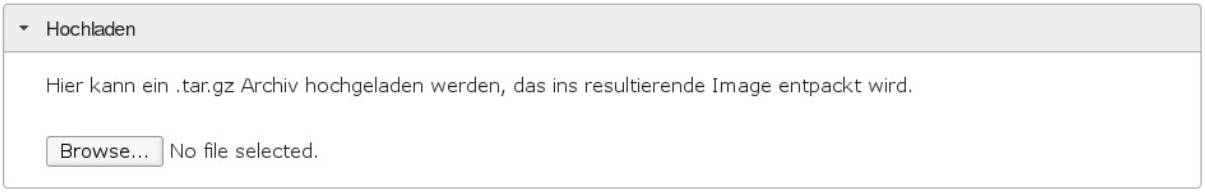
Hier kann die Liste installierter Pakete angepasst werden und so eigene noch benötigte Pakete direkt ins Image gebaut werden. Dies ist auch deshalb praktisch, da direkt ins Image gebaute Pakete dank besserer Kompression weniger Platz beanspruchen als später nachinstallierte Pakete.

Um Pakete zu installieren den Paketnamen im Textfeld hinzufügen. Alternativ können Pakete auch aus der Liste unten durch einen Klick auf das “+”-Symbol hinzugefügt werden.



Upload

Über dieses Formular kann ein .tar.gz Archiv mit eigenen Dateien hochgeladen werden, die ins Firmwareimage kopiert werden sollen. Dadurch lassen sich leicht eigene Anpassungen direkt mit ins resultierende Image packen.



Beispiel: hallowelt.html soll auf dem router in /www erstellt werden. Um diese Datei zu erstellen und zu packen kann man unter Linux so vorgehen:

```
cd /tmp
mkdir www
echo "Hallo Welt" > www/hallowelt.html
tar -cvzf upload.tar.gz www
```

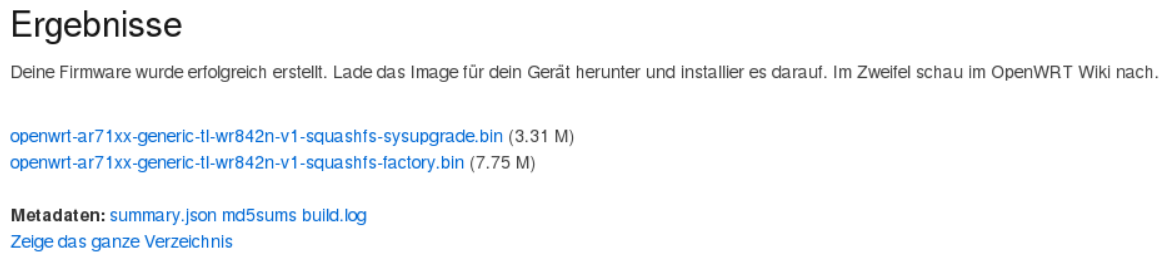
Das Archiv `upload.tar.gz` kann nun über den Button `Browse` ausgewählt und hochgeladen werden. Nach dem Flashen des Routers befindet sie sich auf dem Router im /www Ordner.

Abschicken

Nachdem alle Einstellungen getätigt sind drücke im Tab `Abschicken` noch auf `Submit`. Sind alle Eingaben gültig kommst du zum letzten Schritt des Meshkit, in dem deine Firmware gebaut wird. Bei ungültigen Eingaben weisst Meshkit darauf hin, welche Felder ungültig sind.

3.3.4 3. Firmware erstellen

Die Firmware wird nun erstellt. Das dauert 20s bis einige Minuten. Wenn Meshkit damit fertig ist, zeigt es Links zu den gebauten Images an:



Lade nun die passende Firmwaredatei für den Freifunk Router auf deinen Computer herunter (Rechtsklick -> Speichern unter). oder kopiere den Link (Rechtsklick -> “Link Adresse kopieren” oder dergleichen).

Die eben erstellte Firmware muss dann noch auf den Router geflasht werden: *Meshkit Firmware auf einen Router flashen*

KAPITEL 4

Meshkit Firmware auf einen Router flashen

Nachdem die Meshkit Firmware generiert wurde, muss sie nun noch auf den Router geflasht werden. Bei manchen Routern ist das sehr einfach, bei anderen muss man etwas mehr Aufwand betreiben. Es lohnt sich wie so of auch ein Blick in die OpenWrt Table of Hardware. Dort finden sich oft weitere Angaben zum Flashen bestimmter Modelle.

Inhalt:

4.1 Router flashen auf denen die Originalfirmware installiert ist

Hier wird beschrieben, wie ein Meshkit Firmwareimage auf einen Router geflasht werden kann, auf dem noch die Originalfirmware läuft.

> **Warnung:** Nach dem Flashen sollte als erstes das Passwort des Routers geändert werden, siehe *Passwort ändern*.

Eine Frage die immer wieder auftaucht ist: "Welches Image muss ich benutzen?" Letztendlich verschafft hier nur ein Blick in die Table of Hardware Klarheit. Als Faustregel kann man jedoch sagen:

- Für **ar71xx** muss ein Image mit **factory** im Namen verwendet werden, wenn auf dem Router noch die Originalfirmware läuft.
- Für **brcm47xx** dagegen muss hier in der Regel ein Image mit der Endung **.bin** verwendet werden.

Es gibt sehr viele Router die mit OpenWrt geflasht werden können. Dementsprechend gibt es verschiedene Wege. Manche Router sind leicht zu flashen, z.B. übers Webinterface der Originalfirmware, bei anderen muss man etwas mehr Aufwand betreiben (z.B. Über serielle Konsole und/oder *TFTP*). Aufschluss darüber, wie ein bestimmtes Modell geflasht werden kann gibt, wie so oft, die OpenWrt Table of Hardware.

4.1.1 Firmware übers Webinterface der Originalfirmware flashen

Das ist der einfachste Weg und funktioniert auf vielen Routern. Es kann an dieser Stelle nicht für jedes Modell einzeln erklärt werden, die genau der Router über die oroginale Weboberfläche geflasht werden kann. Das Prinzip ist jedoch bei den meisten Routern ähnlich. Im Folgenden ein paar Beispiele von beliebten Routern mit grosser Verbreitung. Wie immer lohnt auch ein Blick in die Table of Hardware, besonders für hier nicht aufgeführte Modelle.

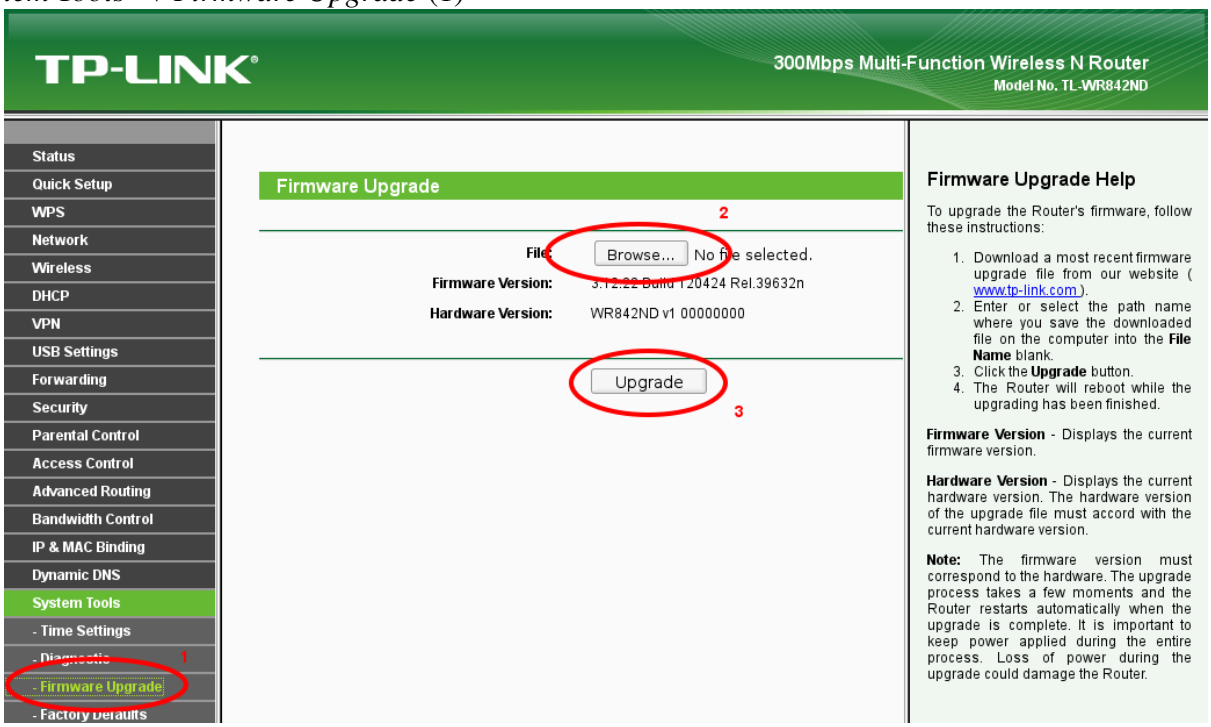
Flashen von TP-Link Routern über das originale Webinterface

TP-Link Router können leicht über das originale Webinterface geflasht werden. Per Default haben sie auf den LAN-Schnittstellen entweder `192.168.0.1` oder `192.168.1.1` als IP-Adresse.

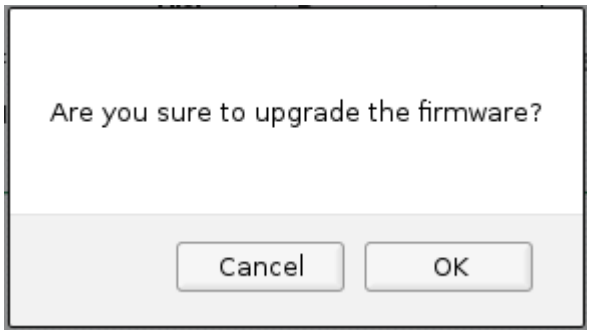
Zunächst musst der PC mit einem der LAN-Ports des Routers verbunden werden, siehe dazu auch *Zugriff auf den Freifunk Router*.

Rufe dann im Browser die Seite http://192.168.0.1 bzw http://192.168.1.1 auf. Dort wirst du zunächst nach Benutzername und Passwort gefragt, welches bei TP-Link beide **admin** sind.

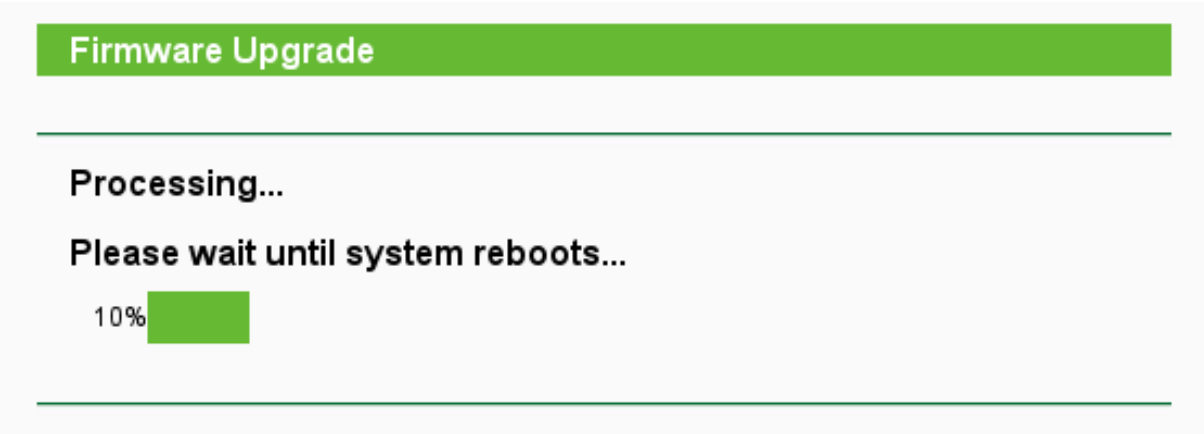
Nach erfolgreichem Login sieht man die originale Weboberfläche des Routers. Wähle hier *System Tools* → *Firmware Upgrade* (1)



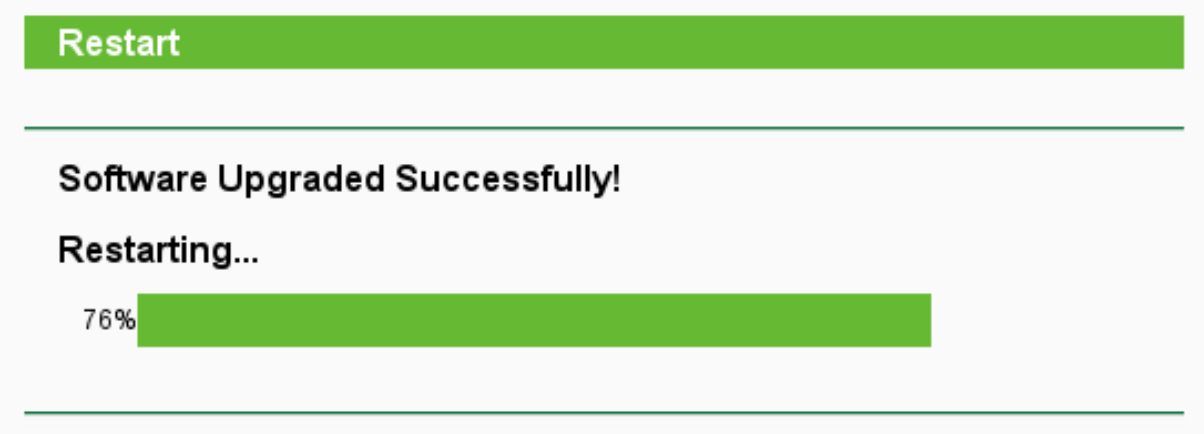
Danach wähle durch einen Klick auf *Browse* (2) die mit Meshkit generierte (siehe *Firmware mit Meshkit generieren lassen*) und auf den eigenen Rechner heruntergeladene Firmware aus. Hier wird ein Image mit **factory** im Namen benötigt. Klicke anschliessend auf *Upgrade* (3). Es kommt eine Nachfrage, ob man sich wirklich sicher ist, die Firmware flashen zu wollen.



Nochmal überlegen... wollen wir natürlich. Also noch eben mit einem Klick auf *OK* bestätigen um den Upgradevorgang zu starten:



War der Flashvorgang erfolgreich, dann erscheint noch



Gut. Die Meshkit Firmware ist nun installiert. OpenWrt verwendet per Default `192.168.1.1` als IP der LAN-Schnittstelle, der automatische Reload der Seite funktioniert als nicht, wenn als IP zum Zugriff auf den Router hier `192.168.0.1` verwendet wurde. Zum Verbinden mit dem frisch geflashten Freifunkrouter siehe *Zugriff auf den Freifunk Router*.

4.2 Router flashen auf dem bereits OpenWrt installiert ist

Hier wird beschrieben, wie ein neues Meshkit Firmwareimage auf einen Router geflasht werden kann, auf dem bereits eine auf OpenWrt basierende Firmware läuft.

Warnung: Nach dem Flashen sollte als erstes das Passwort des Routers geändert werden, siehe *Passwort ändern*.

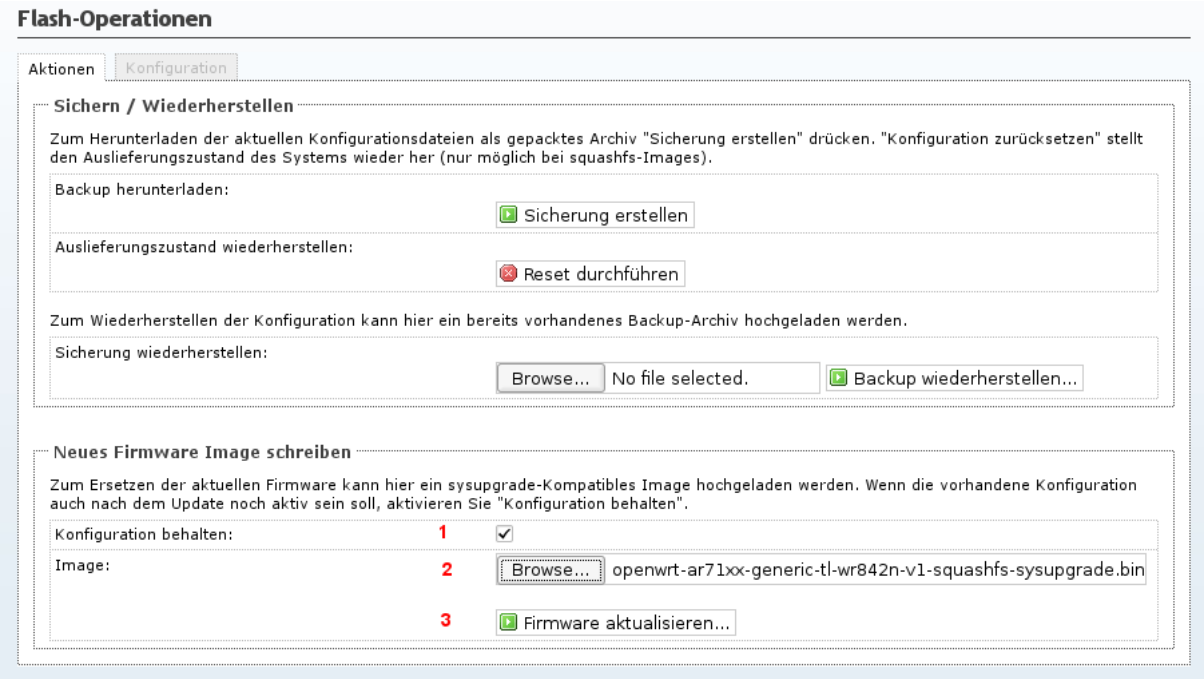
Eine Frage die immer wieder auftaucht ist: "Welches Image muss ich benutzen?" Letztendlich verschafft hier nur ein Blick in die Table of Hardware Klarheit. Als Faustregel kann man jedoch sagen:

- Für **ar71xx** muss ein Image mit **sysupgrade** im Namen verwendet werden, wenn von einem bereits laufenden OpenWrt System aus geflasht wird.
- Für **brcm47xx** dagegen muss hier in der Regel ein Image mit der Endung **.trx** verwendet werden.

4.2.1 Firmware im LuCI Webinterface flashen

Bei den meisten Routern kann Über über das LuCI Webinterface einfach eine neue Firmware installiert werden. Gehe dazu zu

Administration → System → Backup / Firmware Update

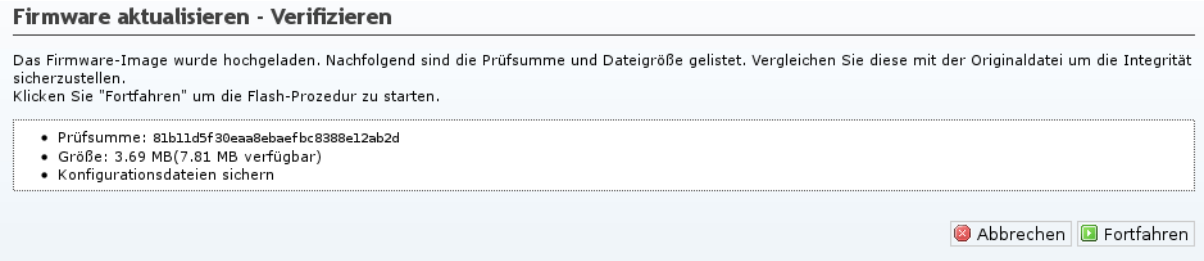


Entferne zunächst den Haken bei *Konfiguration sichern* (1).

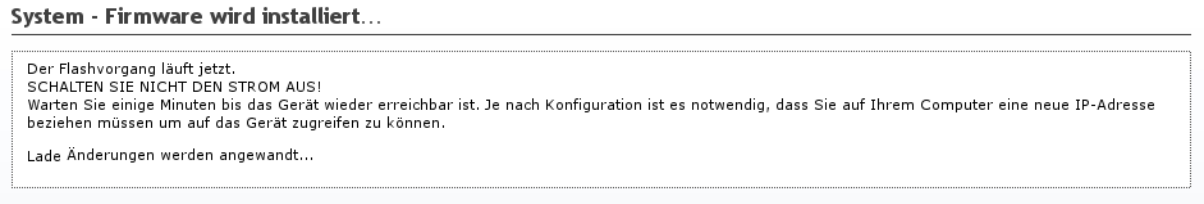
Hinweis: In den meisten Fällen ist es sinnvoll, die bestehende Konfiguration nicht zu sichern und ein komplett frisches Meshkit Firmwareimage zu flashen. Willst du dennoch deine schon bestehende konfiguration sichern, dann wähle beim Generieren des Images mit Meshkit die Option `Keine Konfiguration` (siehe *Erklärung der einzelnen Optionen*) aus und lasse den Haken oben bei *Konfiguration sichern* stehen. Es kann dabei jedoch zu Problemen kommen, z.B. wenn deine bestehende Konfiguration veraltet ist und daher nicht mehr macht, was sie soll.

Klicke anschliessend auf *Browse* (2) und wähle im sich öffnenden Dialog das von Meshkit für deinen Router generierte Firmwareimage (siehe *Firmware mit Meshkit generieren lassen*) aus. Klicke anschliessend auf *Firmware aktualisieren...* (3).

Anschliessend will LuCI nocheinmal eine Bestätigung, dass man das Image wirklich Flashen will. Im Fehlerfall wird eine Meldung über den Fehler der aufgetreten ist ausgegeben. Normalerweise sollte aber die Aufforderung zur Bestätigung des Flashvorgangs erscheinen:



Klicke hier auf *Fortfahren* um mit dem Flashvorgang zu beginnen und folgenden Hinweis zu sehen:



Beachte insbesondere den Hinweis, dass der Router während des Flashens nicht vom Strom getrennt werden sollte.

Das dauert jetzt ein paar Minuten. Hat sich die IP-Adresse des Routers, mit der du verbunden bist nicht geändert, dann sollte nach Beenden des Flashvorgangs die Startseite des Routers neu geladen werden. Ist dies auch nach längerer Zeit nicht der Fall, musst du evtl. die IP-Adresse auf deinem Computer neu einstellen. Siehe dazu *Mit dem Router verbinden*.

4.2.2 Firmware mit sysupgrade auf der Shell flashen

Ist bereits Openwrt installiert, dann kann eine neue Firmware auch sehr einfach über SSH (*Zugang per SSH*) mit **sysupgrade** installiert werden.

Auf der Shell des Routers gibt man dafür ein:

```
cd /tmp
wget <link zum image>
sysupgrade -n <heruntergeladenes image>
```

Hinweis: Hat der Router keine eigene Internetverbindung und **wget** kann das Image daher nicht direkt herunterladen, dann kann ein Image auch mit **scp** nach `/tmp` auf dem Router kopiert werden, siehe *Dateien vom/zum Router kopieren*.

Hinweis: Das **sysupgrade** Kommando sichert durch die Angabe des "-n" Parameters keine Konfiguration. Das ist in der Regel gewollt, da es in den meisten Fällen sinnvoller ist, die bestehende Konfiguration nicht zu sichern und ein komplett frisches Meshkit Firmwareimage zu

flashen. Willst du dennoch deine schon bestehende konfiguration sichern, dann wähle beim Generieren des Images mit Meshkit die Option `Keine Konfiguration` (siehe *Erklärung der einzelnen Optionen*) aus und lasse den "-n" Parameter weg. Es kann dabei jedoch zu Problemen kommen, z.B. wenn deine bestehende Konfiguration veraltet ist und daher nicht mehr macht, was sie soll.

KAPITEL 5

Benutzerhandbuch

5.1 Mit dem Router verbinden

Zu Diagnose- und Konfigurationszwecken kann man sich mit dem Freifunk Router entweder über SSH oder einen Webbrowser verbinden.

5.1.1 Netzwerkverbindung herstellen

Um auf den Router über Netzwerk zuzugreifen brauchst du zunächst eine Netzwerkverbindung mit dem Router. Je nachdem wie dein PC/Notebook mit dem Router verbunden ist (siehe *Einbinden des Freifunkrouters ins eigene Netzwerk*) muss unterschiedlich vorgegangen werden. zunächst muss die IP-Adresse des Routers herausgefunden werden.

Standardsetup: Router mit WAN-Schnittstelle am eigenen LAN

Bemerkung: Dies bezieht sich auf einen Freifunk Router, der so ins eigene Netzwerk eingebunden ist: *Der Standardfall: Freifunk Router am Heimnetzwerk*

Achtung: Zugriffe über die WAN-Schnittstelle des Routers werden normalerweise von der Firewall verboten und müssen explizit erlaubt werden. Am einfachsten geht das indem man im Meshkit im WAN-Tab `Zugriff über SSH erlauben` und `Zugriff über Web erlauben` auswählt. Siehe: *Konfiguration von WAN im Meshkit*.

Bezieht der Freifunkrouter seine IP automatisch per *DHCP* vom eigenen Router, was die Standardeinstellung ist, dann muss zunächst herausgefunden werden, welche IP der Freifunkrouter vom eigenen Router bekommen hat. Dein eigener Router hat hoffentlich ein Webinterface, das irgendwo anzeigt, welche Clients verbunden sind und welche IPs diese haben. Dort solltest du dann die IP des Freifunk Routers finden.

Wurde beim generieren des Images für den Router im Meshkit eine statische IP-Adresse vergeben, dann verwende die dort vergebene IP. Verbinde dich jetzt mit dieser IP zum Freifunkrouter: *Zugriff auf den Freifunk Router*

Freifunk Router direkt am Internet, Client am LAN

Bemerkung: Dies bezieht sich auf einen Freifunk Router, der direkt mit dem Internet verbunden und der eigene PC mit der LAN-Buchse des Freifunk Routers verbunden ist. Siehe: *Freifunk Router direkt am Internetzugang*

Ein PC, der direkt am LAN-Anschluss des Routers angeschlossen ist erhält von diesem automatisch per *DHCP* eine IP-Adresse zugewiesen.

Sofern die IP des LAN-Interfaces des Freifunk Routers nicht verändert wurde ist die IP zum Zugriff auf den Router über die LAN-Schnittstelle immer `192.168.1.1`.

Verbindung über das Freifunknetz (WLAN)

Verbindet man sich per WLAN mit dem Router (ESSID `Freifunk-10.z.x.y` oder `stadt.freifunk.net` erhält man automatisch eine IP vom Router.

Danach kann auch aus dem Freifunknetz kann auf den Router zugegriffen werden. Verwende dazu die IP-Adresse, die du dem Router beim Generieren des Firmwareimages mit dem Meshkit gegeben hast, z.B. 10.0.0.1. Der Router ist unter dieser Adresse aus dem ganzen Freifunknetz erreichbar, d.h. man kann auch auf den Router zugreifen, wenn man an anderer Stelle im Freifunknetz ist.

5.1.2 Zugriff auf den Freifunk Router

Es gibt zwei Wege sich mit dem Router zu verbinden, um Status abzufragen oder die Konfiguration zu ändern. Entweder man öffnet in einem Browser die Webseite des Routers oder verbindet sich per ssh. In beiden Fällen ist das Passwort für den Login `root` und das Defaultpasswort `admin`.

Zugang zum Webinterface

Auf dem Freifunk Router läuft das LuCI-Webinterface, das Statusinformationen anzeigt und wo Einstellungen am Router vorgenommen werden können. Auf das Webinterface kann mit jedem Browser über die Adresse

http://<ip-adresse>

zugegriffen werden.

Zugang per SSH

mit SSH kann man direkt auf die Kommandozeile des Freifunk routers zugreifen und dort mit den gängigen Linux Befehlen arbeiten. Erfahrene Nutzer erledigen Aufgaben auf der Kommandozeile des Routers oft schneller als über das Webinterface.

Bei Linux ist ein SSH-Client meist schon installiert und man kann sich einfach mit

```
ssh root@<ip-adresse>
```

von der Kommandozeile aus mit dem SSH-Server des Freifunkrouters verbinden.

Für Windows ist Putty eine häufig genutzte Anwendung zur Benutzung von SSH.

5.2 Die Shell des Routers

Auf dem Freifunkrouter läuft ja bekanntlich ein OpenWrt System. Hier gibt es viele der üblichen Kommandos die man aus der Linux/Unix Welt kennt. Wer bereits Erfahrung mit der Shell (oder auf Deutsch: Kommandozeile) auf unix-artigen Systemen hat wird sich auf OpenWrt leicht zurechtfinden.

Hinweis: Man muss die Shell nicht unbedingt verwenden. Vieles ist auch über die LuCI Weboberfläche zu bewerkstelligen. Einige Dinge lassen sich aber nur auf der Shell erledigen. Oder zumindest einfacher und schneller.

Im Folgenden sollen einige wichtige und nützliche Kommandos vorgestellt werden.

Hinweis: Die meisten Kommandos haben eine eingebaute Hilfe die angezeigt wird, wenn man dem Kommando `-h` oder `--help` anhängt.

5.2.1 Bewegen im Dateisystem

Wie unter Linux üblich gibt es ein Rootverzeichnis, das `/` heisst. Alle weiteren Verzeichnisse sind Unterverzeichnisse davon.

Den Inhalt des aktuellen Ordners kann man mit **ls** anzeigen.

Um sich im Dateisystem zu bewegen wird **cd** verwendet:

Kommando	Beschreibung
cd ..	Eine Ebene nach oben wechseln
cd /	Ins Rootverzeichnis wechseln
cd <verzeichnis>	Ins Verzeichnis <verzeichnis> wechseln, z.B.
cd /etc/config	wechselt ins Verzeichis `/etc/config`

5.2.2 Dateien anzeigen und editieren

Standardmässig ist **vi** als Editor auf dem Router installiert. Für Hilfe zur Benutzung von **vi** siehe:

Dateien bearbeiten im OpenWrt Wiki

Als Alternative zu **vi** ist auch **nano** empfehlenswert. Dieser ist nicht per Default installiert sondern muss nachinstalliert werden. Entweder direkt von Meshkit mit ins Firmwareimage bauen lassen (siehe *Pakete*) oder nachinstallieren (*Pakete installieren*).

Will man eine **Datei nur anzeigen**, dann kann hierfür auch **cat** verwendet werden, z.B. **cat /etc/banner**.

5.2.3 uci

Die Konfiguration von OpenWrt erfolgt in der Regel über Konfigurationsdateien in `/etc/config`, die das `Unified Configuration` Format verwenden. Diese Dateien können direkt bearbeitet werden. Es ist aber auch möglich, **uci** zu verwenden, um Einstellungen in diesen Files zu verändern.

Kommando	Beschreibung
uci show	zeigt alle uci Konfigurationseinstellungen
uci show freifunk	zeigt alle Einträge in `/etc/config/freifunk`
uci show freifunk.contact	zeigt alle Einträge der Sektion `contact` in `/etc/config/freifunk`
uci get freifunk.contact.nickname	Zeigt die Option `nickname` aus der Section `contact` in `/etc/config/freifunk`
uci set freifunk.contact.nickname='freifunker'	Setzt die Option `nickname` aus der Section `contact` in `/etc/config/freifunk` auf 'freifunker'
uci changes	zeigt alle gemachten, aber noch nicht committeten Änderungen
uci commit	schreibt die gemachten Änderungen in allen Konfigurationsdateien
uci –help	Hilfe anzeigen

5.2.4 Netzwerkprobleme debuggen

nslookup

Sagt uns ob die Namensauflösung für einen Rechner funktioniert.

Beispiel:

```
root@freifunk:~# nslookup google.com
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      google.com
Address 1: 2a00:1450:4001:c02::65 fa-in-x65.1e100.net
Address 2: 173.194.70.101 fa-in-f101.1e100.net
```

aber:

```
root@freifunk:~# nslookup domaindieesnichtgibt.de
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

nslookup: can't resolve 'domaindieesnichtgibt.de': Name or service not known
```

ping und ping6

Sendet ICMP echo requests an einen Rechner und zeigt die Paketlaufzeit, sofern der `angepingte` Rechner antwortet.

Beispiel:

```
root@freifunk:~# ping google.de
PING google.de (173.194.32.255): 56 data bytes
64 bytes from 173.194.32.255: seq=0 ttl=52 time=51.953 ms
64 bytes from 173.194.32.255: seq=1 ttl=52 time=51.293 ms
[...]
```

Für IPv6 verwendet man dementsprechend **ping6**:

```
root@freifunk:~# ping6 google.de
PING google.de (2a00:1450:4016:803::1017): 56 data bytes
64 bytes from 2a00:1450:4016:803::1017: seq=0 ttl=55 time=135.781 ms
64 bytes from 2a00:1450:4016:803::1017: seq=1 ttl=55 time=132.964 ms
```

Bekommen wir eine Antwort vom Rechner, dann funktioniert die Kommunikation mit ihm. Wenn nicht dann kann uns *traceroute und traceroute6* unter Umständen sagen, wo das Problem liegt.

traceroute und traceroute6

Mit **traceroute** kann die Route eines Pakets durchs Netz verfolgt werden.

Beispiel:

```
root@freifunk:~# traceroute openlab-indoor.ffa
traceroute to openlab-indoor.ffa (10.11.0.20), 30 hops max, 38 byte packets
1  kerberos.ffa (10.11.0.17)  4.165 ms  2.102 ms  1.234 ms
2  mid1.hirsch.ffa (10.11.63.22)  66.282 ms  62.849 ms  61.505 ms
3  openlab-balkon.ffa (10.11.0.21)  73.504 ms  94.929 ms  86.264 ms
4  openlab-indoor.ffa (10.11.0.20)  97.610 ms  83.681 ms  78.941 ms
```

Hier wird die Route zu einem anderen Knoten im Mesh-Netzwerk. Das Paket durchläuft die Router 1, 2, 3 um schliesslich bei 4 das Ziel zu erreichen.

Für IPv6 wird **traceroute6** verwendet, das im Paket `iputils-traceroute6` enthalten ist und u.U. erst noch installiert werden muss, siehe *Pakete installieren*.

5.2.5 iproute2

ip ist ein mächtiges Kommando zum Anzeigen und Verändern von IP-Konfiguration wie IP-Adressen und Routingeinträge. Es sollte dem älteren **ifconfig** vorgezogen werden.

Allgemeines:

- durch den Parameter "-4" werden nur IPv4-Informationen angezeigt, durch "-6" ur IPv6-Informationen

- Alle Kommandos können auch in Kurzschreibweise angegeben werden, z.b. **ip a** statt **ip addr**

Wichtige Kommandos sind:

Kommando	Beschreibung
ip addr	Zeigt die Konfiguration aller Netzwerkschnittstellen an
ip -4 addr show dev eth1	zeigt alle IPv4-Informationen zur Schnittstelle `eth1` an.
ip addr add 1.2.3.4/32 dev eth1	Konfiguriert `1.2.3.4` mit der Netzmaske `32` als (weitere) IP-Adresse für `eth1`.
ip addr del 1.2.3.4/32 dev eth1	löscht die oben angelegte Adresse `1.2.3.4` auf `eth1` wieder.
ip route show	Zeigt alle Routen in der `main`-Routingtabelle.
ip route show table olsr	Zeigt alle Routen in der `olsr`-Routingtabelle.
ip route add 192.168.100.0/24 via 192.168.2.1 dev eth1	Fügt eine Route zum Netzwerk `192.168.100.0/24` in die `main`-Routingtable ein. Der Gateway für dieses Netzwerk ist `192.168.2.1` und das Interface dafür `eth1`.
ip route del 192.168.100.0/24 via 192.168.2.1 dev eth1	Löscht die oben angelegte Route zu `192.168.100.0/24` wieder.
ip route add 192.168.100.0/24 via 192.168.2.1 dev eth1 table olsr	Fügt eine Route zum Netzwerk `192.168.100.0/24` in die `olsr`-Routingtable ein. Der Gateway für dieses Netzwerk ist `192.168.2.1` und das Interface dafür `eth1`.
ip route del 192.168.100.0/24 via 192.168.2.1 dev eth1 table olsr	Löscht die oben angelegte Route zu `192.168.100.0/24` wieder.
ip rule show	zeigt alle Routingregeln ("ip rules") an
ip –help	zeigt die Hilfe zu **ip** an, das noch viel mehr kann als hier gezeigt.

5.2.6 ipcalc.sh

Ist ein nützliches kleines Kommando auf der OpenWrt Shell. Insbesondere kann damit die Startadresse (`NETWORK`) sowie die Endadresse (`BROADCAST`) eines gegebenen Netzwerks berechnet werden. Die Netzmaske wird sowohl in `dotted decimal notation` (`NETMASK`) als auch in der `CIDR`-Schreibweise (`PREFIX`) auegebenen. Damit ist **ipcalc.sh** auch sehr gut geeignet, um Netzmasken von der einen in die andere Notation umzuwandeln.

Beispiel: Angabe des Netzwerks in der `CIDR`-Notation:

```
root@freifunk:~# ipcalc.sh  10.11.0.1/18
IP=10.11.0.1
NETMASK=255.255.192.0
BROADCAST=10.11.63.255
NETWORK=10.11.0.0
PREFIX=18
```

Beispiel: Angabe des Netzwerks in der `Dotted Decimal`-Notation:

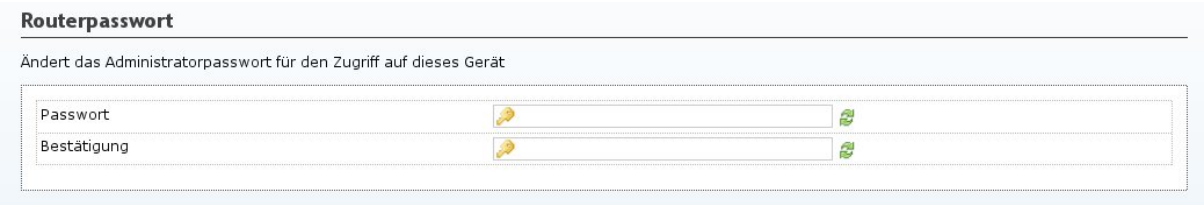
```
root@freifunk:~# ipcalc.sh  10.11.0.1 255.255.192.0
IP=10.11.0.1
NETMASK=255.255.192.0
BROADCAST=10.11.63.255
NETWORK=10.11.0.0
PREFIX=18
```

5.3 Passwort ändern

Setzt voraus: *Mit dem Router verbinden*

5.3.1 Im Webinterface

Das Passwort kann unter *Administration* → *System* → *Administration* geändert werden.



5.3.2 Mit SSH

Auf dem Shell des Routers **passwd** eingeben. Dann das neue Passwort eingeben und noch einmal bestätigen.

```
root@freifunk:~# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@freifunk:~#
```

5.4 Grundeinstellungen

Zu finden unter *Administration* → *Freifunk* → *Grundeinstellungen*

5.4.1 Community

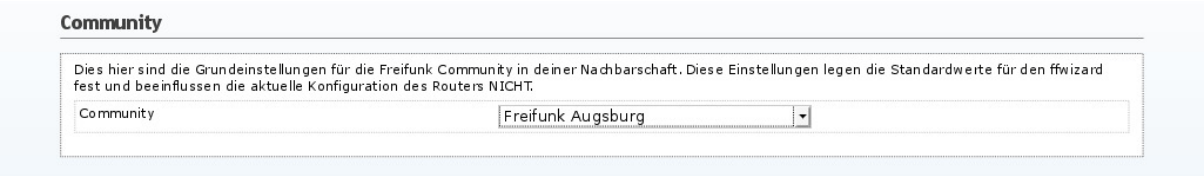
Wählt die Community. Dadurch werden Default-Werte für den Meshwizard vorgegeben. In der Regel gibt es hier nichts auszuwählen. Soll ein Router in einer anderen Community neu in Betrieb genommen werden, dann ist es sinnvoller, für diesen ein neues Image mit Meshkit zu erstellen und den Router neu zu flashen.

Will man es dennoch ohne neu zu flashen versuchen, dann kann man mit

```
opkg update
opkg install community-profiles
```

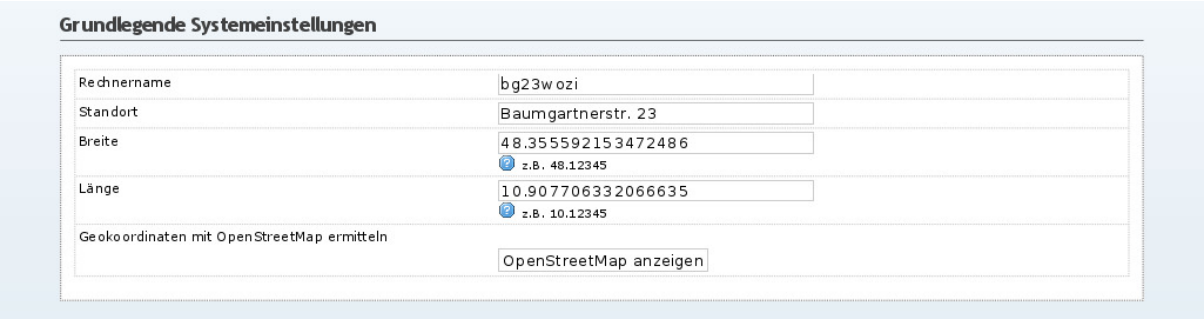
alle weiteren existierenden Community-Profile installieren.

Siehe auch: *Community Profile*



5.4.2 Grundlegende Systemeinstellungen

Hier können der **Name des Routers**, der **Standort** sowie die **Geokoordinaten** geändert werden. Ein Klick auf *OpenStreetmap anzeigen* öffnet eine Karte, in der die Geokoordinaten bequem ausgewählt werden können.



5.5 Kontakt

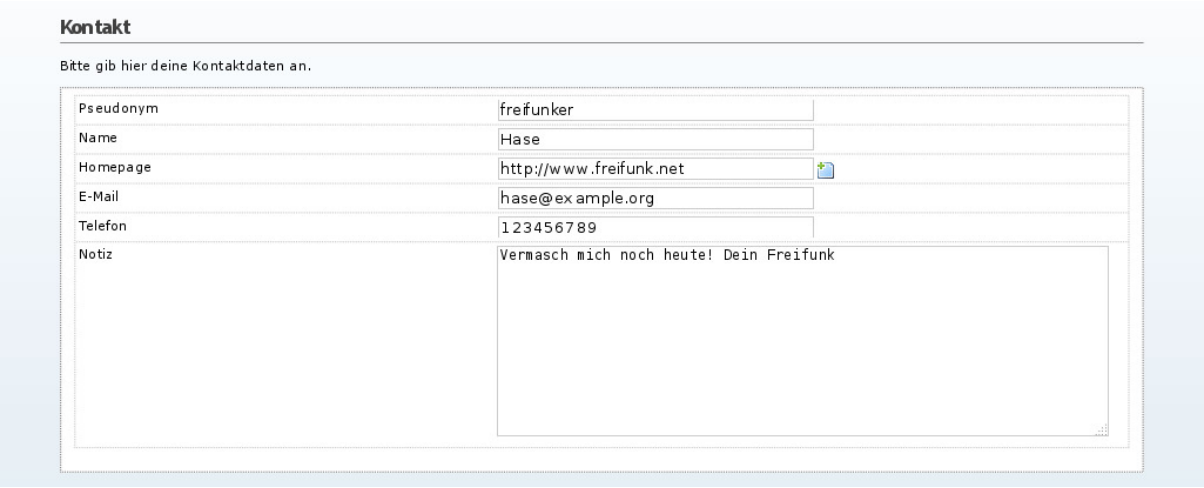
Kontaktdaten können entweder im LuCI-Webinterface oder auf der Shell geändert werden.

Hinweis: Es ist empfehlenswert zumindest einen Nickname und eine gültige Emailadresse anzugeben, damit interessierte Nutzer oder andere Betreiber von Freifunkknoten Kontakt aufnehmen können.

5.5.1 LuCI

Zu finden unter *Administration* → *Freifunk* → *Kontakt*

Hier können Kontaktdaten wie ein **Pseudonym** (dein Nickname), **Name**, **Telefonnummer**, **Email Adresse** und **Homepage** angegeben werden, die dann im öffentlichen Teil des Webinterfaces unter *Freifunk* → *Kontakt* angezeigt werden. Zudem kann eine kurze Notiz hinterlegt werden.



5.5.2 Auf der Shell

Entweder durch direktes Bearbeiten von `/etc/config/freifunk`:

```
config public 'contact'
        option phone '123456789'
        option note 'Vermasch mich noch heute! Dein Freifunk'
        option nickname 'freifunker'
        option name 'Hase'
        list homepage 'http://www.freifunk.net'
        option mail 'hase@example.org'
```

oder über **uci** Kommandos:

```
uci set freifunk.contact.phone=123456789
uci set freifunk.contact.note=Vermasch mich noch heute! Dein Freifunk
uci set freifunk.contact.nickname=freifunker
uci set freifunk.contact.name=Hase
uci set freifunk.contact.homepage=http://www.freifunk.net
uci set freifunk.contact.mail=hase@example.org
uci commit freifunk
```

5.6 Pakete installieren

Auf OpenWrt-Systemen können weitere Pakete nachinstalliert werden, um den Funktionsumfang des Routers zu vergrössern und um bestimmte Aufgaben zu erfüllen.

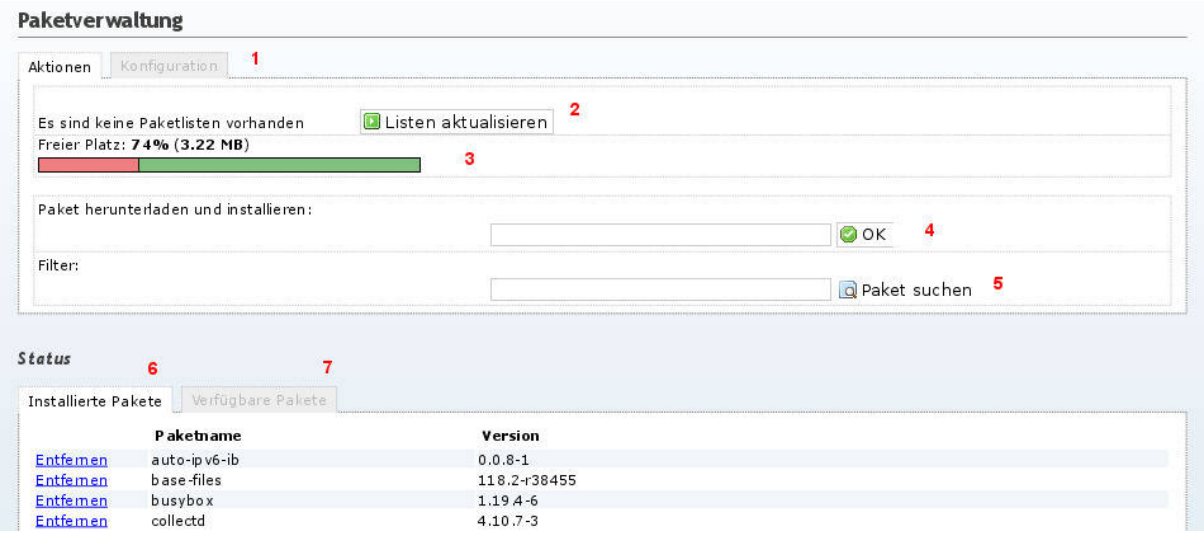
Die Pakete werden durch `Repositories` bereitgestellt, die in `/etc/opkg.conf` festgelegt sind. Jedes Repository enthält eine Datei `Packages`, die eine Liste aller dort verfügbaren Pakete und Informationen zu diesen (Grösse, Beschreibung, Abhängigkeiten...) enthält. Diese Listen können relativ gross sein und werden deshalb nicht permanent auf dem Router gespeichert. **Deshalb müssen, bevor Pakete installiert werden können, die Paketkisten aktualisiert werden**.

Hinweis: Wenn man schon im voraus weiss, welche Pakete man zusätzlich zu den Standardpaketen auf dem Router benötigt dann ist es sinnvoll, diese gleich von Meshkit mit ins Firmwareimage bauen zu lassen, da dadurch durch bessere Kompression Platz im Flash-Speicher des Routers gespart werden kann. Siehe: *Pakete*.

Hinweis: Der Router braucht eine Internetverbindung damit Pakete installiert werden können.

5.6.1 Im Webinterface

Pakete lassen sich in LuCI unter *Administration* → *System* → *Paketverwaltung* verwalten.



Im Tabmenü bei (1) kann *Aktionen* oder *Konfiguration* gewählt werden. Unter Konfiguration kann die opkg-Konfiguration angepasst werden. Darauf gehen wir hier jedoch nicht weiter ein.

(3) zeigt den noch verfpgbaren Platz im Flash Speicher des Routers an, der noch für die Installation von Paketen zur Verfügung steht.

Pakete installieren

Zunächst müssen, sofern noch nicht geschehen, die Paketlisten durch einen Klick auf *Listen aktualisieren* (2) heruntergeladen werden. Danach können Pakete auf verschiedene Art installiert werden:

- Durch direkte Angabe eines Paketnamens unter *Paket herunterladen und installieren* (4).
- Indem man nach einem Paket unter *Filter* sucht (5).
- Durch Auswählen aus der Liste unten im Tab *Verfügbare Pakete* (7)

Pakete deinstallieren

Durch KLicken auf *Entfernen* im Tab *Installierte Pakete* (6) können einzelne Pakete entfernt werden.

5.6.2 Mit SSH

Das Kommando zum Arbeiten mit Paketen heisst **opkg**.

Aktion	Kommando
Hilfe anzeigen	opkg -h
Paketlisten updaten	opkg update
Paket suchen	opkg find *<Suchbegriff>*
Info zu einem paket anzeigen	opkg info <Paketname>
Paket installieren	opkg install <Paketname>
Paket entfernen	opkg remove <Paketname>
Installierte Pakete anzeigen	opkg list_installed

Werden Dienste (wie z.B. ein FTP-Server) installiert, die durch ein init-Script in `/etc/init.d/` gestartet werden dann muss das betreffende init-Script noch aktiviert und der Dienst danach gestartet werden. Als Beispiel hier für den FTP-Server `vsftpd`:

```
opkg update
opkg install vsftpd
/etc/init.d/vsftpd enable
/etc/init.d/vsftpd start
```

Offlineinstallation von Paketen

Hat der Router keine Internetverbindung dann können Pakete installiert werden, indem sie in den `/tmp`-Ordner des Routers kopiert werden und dann mit **opkg install /tmp/<paketname>** installiert werden. Zum Kopieren von Dateien auf den Router siehe: *Dateien vom/zum Router kopieren*.

5.6.3 Weiterführende Links

- opkg im OpenWrt Wiki
- packages im OpenWrt Wiki

5.7 Logs anzeigen

Auf dem Router gibt es zwei wichtige Logs, die wichtige Informationen zum System geben können: `System Log` und `Boot log`. Die Logs sind insbesondere bei der Fehlersuche oft hilfreich.

5.7.1 System Log

Das System Log enthält alle Ereignisse, deren Priorität mindestens so hoch ist wie der eingestellte `LogLevel`. Das System Log wird in einen `Ring Buffer` geschrieben, d.h. es steht

nur eine begrenzte Anzahl an Speicherplatz fürs Log zur Verfügung. Wird dieser überschritten, dann werden die ältesten Log-Meldungen verworfen um Platz für neue zu schaffen.

Im **LuCI Webinterface** befindet sich das System Log unter *Administration* → *Status* → *Systemprotokoll*.

Auf der Shell kann das System Log mit dem Kommando **logread** angezeigt werden.

5.7.2 Kernel Log (Dmesg)

Das Log des Kernels kann in LuCI unter *Administration* → *Status* → *Kernelprotokoll* angezeigt werden.

Auf der Shell kann das Kernel-Log mit dem Kommando **dmesg** angezeigt werden.

5.8 Logging konfigurieren

Die Defaulteinstellungen für das Loggen von Ereignissen sind relativ gut und müssen nicht angepasst werden. Dennoch ist es möglich die Konfiguration der Loggingmechanismen an besondere Bedürfnisse anzupassen. Folgende Einstellungen können gemacht werden:

Option LuCI	Option Shell	Beschreibung
Größe des Systemprotokoll-Puffers	buffersize	Größe des `Ring Buffer` in kByte, in dem das Log gespeichert wird.
Externer Protokollserver IP	log_ip	Optional: IP-Adresse eines externen Syslog-Servers, zu dem die Logeinträge geschickt werden
Externer Protokollserver Port	log_port	Optional: Port des externen Syslog-Servers
Protokolllevel	conloglevel	Logmeldungen müssen mindestens diese Priorität haben, um geloggt zu werden. Siehe *Loglevel für Syslog-Meldungen*.
Cron Protokolllevel	cronloglevel	Cronmeldungen müssen mindestens diese Priorität haben, um geloggt zu werden. Siehe *Loglevel für Cron-Meldungen*.

5.8.1 Mögliche Loglevel

Nachrichten werden nur geloggt, wenn deren Priorität mindestens dem konfigurierten Level entspricht.

Loglevel für Syslog-Meldungen

Gibt an, ab welchem Level Meldungen von *cron* ins Syslog geschrieben werden.

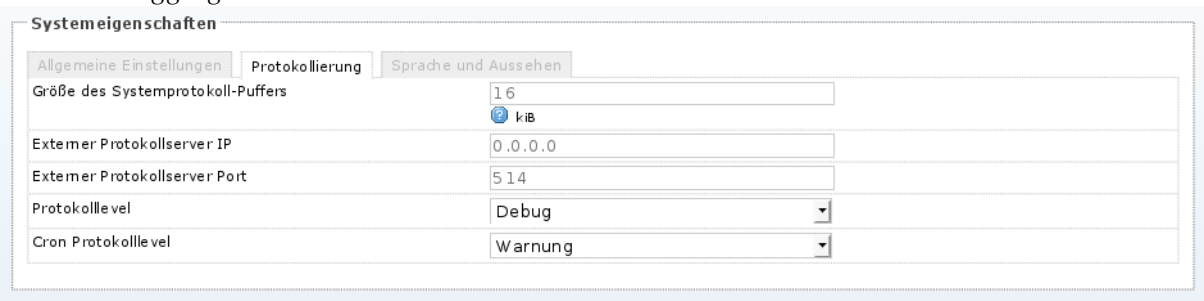
Level	in LuCI	numerisch	Beschreibung
LOG_EMERG	Notfall	1	Das System ist unbenutzbar
LOG_ALERT	Alarm	2	Es ist unbedingt ein sofortiges Eingreifen notwendig
LOG_CRIT	Kritisch	3	Kritische Warnungen
LOG_ERR	Fehler	4	Ein Fehler ist aufgetreten
LOG_WARNING	Warnung	5	Eine Warnung
LOG_NOTICE	Notiz	6	Wichtige Hinweise anzeigen
LOG_INFO	Info	7	rein informelle Ausgaben
LOG_DEBUG	DEBUG	8	ALLE Meldungen werden ausgegeben

Loglevel für Cron-Meldungen

in LuCI	numerisch	Beschreibung
Warnung	9	Nur Fehler von Cron werden gelogged
Normal	8	Normales Logging von Cron-Programmaufrufen
Debug	5	Debug-Meldungen ausgeben

5.8.2 Logging konfigurieren in LuCI

Gehe zu *Administration* → *System* → *System* und wechsle im Abschnitt *Systemeigenschaften* zum Tab *Logging*.



Zur Konfiguration der einzelnen Optionen siehe *Einstellungen fürs Syslog*.

Hinweis: Diese Einstellungen werden erst nach einem Neustart des Systems aktiv.

5.8.3 Logging konfigurieren auf der Shell

Die Konfiguration fürs die Logs befindet sich in `/etc/config/system` in der Sektion `system`. Sie kann dort direkt bearbeitet werden. Alternativ können einzelne Optionen auch mit **uci** verändert werden:

```
uci set system.system.buffersize=16
uci set system.system.cronloglevel=9
uci set system.system.conloglevel=8
uci set system.system.log_ip='1.2.3.4'
```

```
uci set system.system.log_port=514
uci commit system
```

Zur Konfiguration der einzelnen Optionen siehe *Einstellungen fürs Syslog*.

Hinweis: Diese Einstellungen werden erst nach einem Neustart des Systems aktiv.

5.9 DHCP-Splash

Der DHCP-Splash ist ein Captive Portal, mit dem Gästen im Freifunknetz beim ersten Zugriff aufs Internet eine **Begrüssungsseite** angezeit werden kann.

Splash kann ausserdem die **Bandbreite im Up- und Download pro Nutzer limitieren**. Zudem ist es möglich, einzelne Clients dauerhaft zu erlauben (*Whitelist*) oder dauerhaft zu sperren (*Blacklist*).

Gründe Splash einzusetzen sind

- Nutzer auf Freifunk aufmerksam machen und zum mitmachen auffordern
- Nutzer über Risiken aufklären (Netzwerk ist unverschlüsselt)
- Nutzungsbedingungen anzeigen und den Nutzer akzeptieren lassen. "Mach nix verbotenes". Dies ist in rechtlicher Hinsicht sinnvoll.
- Indem man durch den Splash eine regelmässige Zwangstrennung hat wird das Freifunknetz für Filesharer eher uninteressant.
- Hinweis auf den Knotenbetreiber
- Lokalitätsbezogene Informationen anzeigen
- Manche nennen den Splash auch "Nervseite". Indem reine Nutzer mit dem Splash genervt werden, sollen sie ermutigt werden, sich aktiv am Netz zu beteiligen.

Gegen einen Einsatz von Splash spricht

- Nutzer werden genervt. Man kann dies aber auch wie oben als Vorteil sehen.
- Aus technischen Gründen funktioniert nur eine Umleitung von HTTP-Requests. Daher können sowohl HTTPS als auch alle anderen Protokolle nicht aufs Internet zugreifen, solange der Splash nicht akzeptiert wurde indem zumindest einmal eine HTTP-Seite aufgerufen wird.

Warnung: Splash darf nicht andere aktive Teilnehmer/Knoten im Freifunknetz behindern. Daher Splash nur auf Schnittstellen betreiben, die nur für Clients (Gäste) benutzt werden.

5.9.1 Splash Status anzeigen

Statusinformationen zum Splash (verbundene Clients, Blacklist, Whitelist) können entweder in LuCi oder auf der Shell angezeigt werden.

Splash Status in LuCI anzeigen

Gehe zu *Administration* → *Status* → *Client-Splash*. Es werden alle derzeit verbundenen Clients angezeigt:

Richtlinie zeigt den aktuellen Status des Clients. Dieser kann dort auch geändert werden.

Richtlinie	Beschreibung
erlaubt	Client ist auf der Whitelist.
gesplasht	Client hat den Splash akzeptiert und ist freigeschalten
vorübergehend geblockt	Lease des Clients entfernen. Er kann sich jedoch erneut freischalten.
gesperrt	Client dauerhaft auf die Blacklist setzen. Er kann den Splash dann nicht mehr akzeptieren und bekommt stattdessen eine Meldung angezeit, dass er geblockt wurde.

Hinweis: Statusinformationen zu Splash-Clients sind in teilweise anonymisierter Form auch im öffentlichen Teil des Webinterfaces sichtbar: *Freifunk* → *Status* → *Splash*.

Hinweis: Mit dem Paket `collectd-mod-splash-leases` können Grafiken über verbundene Clients erstellt werden.

Splash Status auf der Shell anzeigen

Informationen zu verbundenen Clients erhält man auf der Shell mit:

```
luci-splash list
```

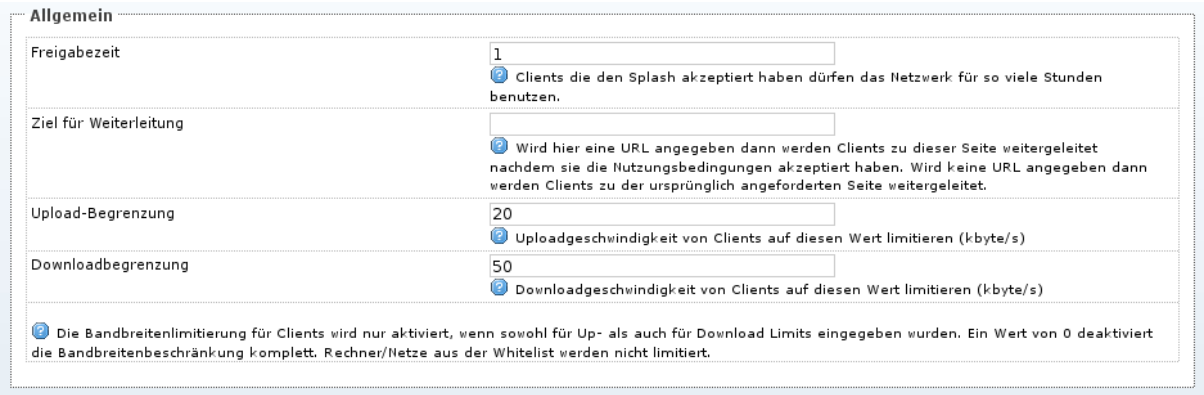
5.9.2 Allgemeine Einstellungen für Splash

Hier können die Freigabezeit, Up-/Downloadlimit und ein Weiterleitungsziel eingerichtet werden.

Option LuCI	Option Shell	Beschreibung
Freigabezeit	leasetime	Die Freigabezeit in Stunden. So lange kann der Gast das Netz benutzen, bevor er den Splash erneut akzeptieren muss.
Ziel für Weiterleitung	redirect_url	Auf diese Seite wird der Nutzer nach Akzeptieren des Splashs weitergeleitet. Wird die Option leer gelassen, dann wird der Nutzer direkt auf die Seite weitergeleitet, auf die er ursprünglich zugreifen wollte.
Upload-Begrenzung	limit_up	Upload-Limit in KByte pro Sekunde. Die Limitierung gilt pro Client. Ein Wert von 0 deaktiviert die Begrenzung. Clients die auf der *Whitelist* stehen sind von der Begrenzung ausgenommen.
Download-Begrenzung	limit_up	Download-Limit in KByte pro Sekunde. Die Limitierung gilt pro Client. Ein Wert von 0 deaktiviert die Begrenzung. Clients die auf der *Whitelist* stehen sind von der Begrenzung ausgenommen.

In LuCI

Öffne *Administration* → *Dienste* → *Client-Splash*. Ganz oben siehst du gleich die allgemeinen Einstellungen für Splash:



Auf der Shell

Um diese allgemeinen Einstellungen auf der Shell vorzunehmen kann entweder `/etc/config/luci_splash` direkt bearbeitet werden:

```
config core 'general'
     option leasetime '1'
     option limit_up '20'
     option limit_down '50'
     option redirect_url 'http://www.freifunk.net'
```

oder die selben Einstellungen mit **uci** gemacht werden:

```
uci set luci_splash.general.leasetime=1
uci set luci_splash.general.limit_up=20
uci set luci_splash.general.limit_down=50
uci set luci_splash.general.redirect_url='http://www.freifunk.net'
uci commit luci_splash
```

Anchliessend muss Splash mit:

```
/etc/init.d/luci_splash
```

neu gestartet werden damit die Änderungen wirksam werden.

5.9.3 Interfaces zum Splash hinzufügen

Um ein Interface zum Splash hinzuzufügen (damit also Clients über dieses Interface den Splash akzeptieren müssen), muss der Name des Netzwerks sowie die Firewallzone zu der das Netzwerk gehört bekannt sein.

Interface unter LuCI hinzufügen

Öffne *Administration* → *Dienste* → *Client-Splash*. In der Interface-Sektion sieht man bereits Konfigurierte Schnittstellen:

Durch einen Klick auf *Hinzufügen* können weitere Schnittstellen hinzugefügt werden.

Interface auf der Shell hinzufügen

Interfaces die Splash benutzen soll werden konfigguriert in `/etc/config/luci_splash`. Um ein neues Interface zum Splash hinzuzufügen kann dort direkt am Ende eine neue interface-Sektion eingefügt werden:

```
config iface 'wireless0custom'
      option network 'wireless0custom'
      option zone 'freifunk'
```

Alternativ kann dieser Eintrag auch mit **uci** erstellt werden:

```
uci set luci_splash.wireless0custom=iface
uci set luci_splash.wireless0custom.network=wireless0custom
uci set luci_splash.wireless0custom.zone=freifunk
uci commit luci_splash
```

In beiden Fällen muss anschliessend der Splash mit **/etc/init.d/luci_splash restart** neu gestartet werden, damit die Änderungen wirksam werden.

5.9.4 Whitelist - Clients dauerhaft erlauben

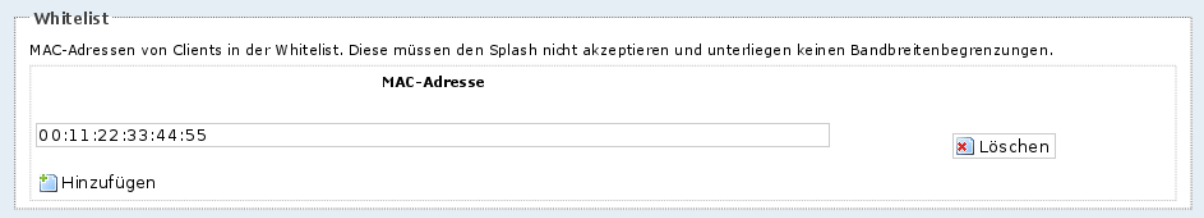
Clients deren *MAC*-Adresse auf der Whitelist steht werden dauerhaft freigeschaltet, d.h. sie müssen nicht den Splash akzeptieren bevor sie ins Internet dürfen. Ausserdem unterliegen sie nicht dem Bandbreitenlimit für normale Clients falls die Bandbreite für diese limitiert wird (siehe *Allgemeine Einstellungen für Splash*).

Um Clients zur Whitelist hinzuzufügen wird deren *MAC*-Adresse benötigt. Clients die verbunden sind können auch anhand ihrer IP-Adresse zur Whitelist hinzugefügt werden.

Clients whitelisten in LuCI

Clients können auf der Statusseite von Splash (siehe *Splash Status in LuCI anzeigen* gewhitelistet werden.

Alternativ ist dies auch möglich über die Splash Einstellungen unter *Administration* → *Dienste* → *Client-Splash*.



Clients whitelisten auf der Shell

Um Clients auf der Shell auf die Whitelist zu setzen:

```
luci-splash whitelist 00:11:22:33:44:55
```

Ist der Client verbunden und die IP-Adresse bekannt (siehe *Splash Status auf der Shell anzeigen*) dann kann der Client auch anhand der IP freigeschalten werden:

```
luci-splash whitelist 1.2.3.4
```

5.9.5 Blacklist - Clients dauerhaft sperren

Clients deren *MAC*-Adresse auf der Blacklist steht werden dauerhaft gesperrt, d.h. sie können den Splash nicht mehr akzeptieren und bekommen stattdessen eine Hinweisseite, dass sie geblockt wurden.

Um Clients zur Blacklist hinzuzufügen wird deren *MAC*-Adresse benötigt. Clients die verbunden sind können auch anhand ihrer IP-Adresse zur Blacklist hinzugefügt werden.

Clients blacklisten in LuCI

Clients können auf der Statusseite von Splash (siehe *Splash Status in LuCI anzeigen* geblacklistet werden.

Alternativ ist dies auch möglich über die Splash Einstellungen unter *Administration* → *Dienste* → *Client-Splash*.

Clients blacklisten auf der Shell

Um Clients auf der Shell auf die Blacklist zu setzen:

```
luci-splash blacklist 00:11:22:33:44:55
```

Ist der Client verbunden und die IP-Adresse bekannt (siehe *Splash Status auf der Shell anzeigen*) dann kann der Client auch anhand der IP gesperrt werden:

```
luci-splash blacklist 1.2.3.4
```

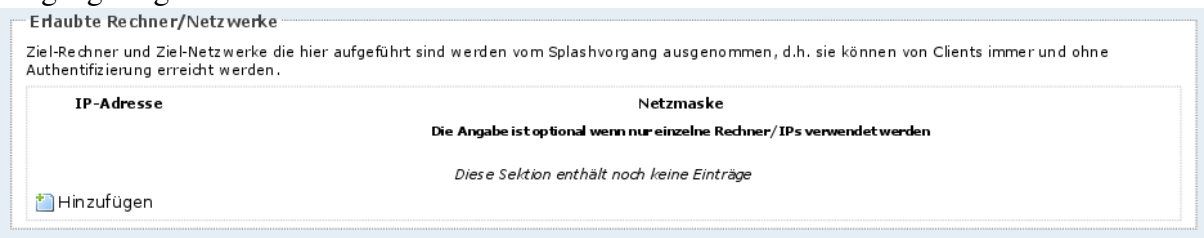
5.9.6 Bestimmte Ziele allgemein erlauben

Einzelne Zielrechner bzw -netzwerke können generell erlaubt werden. Verbindungen zu diesen sind immer möglich, es muss nicht zuerst der Splash erlaubt werden.

Hinweis: Zugriff auf die im Communityprofil definierte **Community-Homepage** (siehe *Community Profile*) wird automatisch erlaubt. Der Server, der diese Webseite bereitstellt muss hier also nicht extra eingetragen werden.

In LuCI

Gehe zu *Administration* → *Dienste* → *Client-Splash*. Dort kannst du unter *Erlaubte Rechner/Netzwerke* durch einen Klick auf *Hinzufügen* Ziel-Netzwerke hinzufügen, die vom Splashvorgang ausgenommen sein sollen.



Soll nur ein einzelnen Zielrechner erlaubt werden, dann reicht es, dessen IP-Adresse anzugeben. Um ein ganzes Ziel-Netzwerk zu erlauben ist zusätzlich die Angabe einer *Netzmaske* in der dotted decimal Schreibweise (z.B. 255.255.255.0) notwendig.

Speichere deine Änderungen mit *Speichern & Anwenden*.

Auf der Shell

Die Einstellungen sind in der Datei `/etc/config/luci_splash` gespeichert. Um ein weiteres Ziel-Netzwerk, wir wollen es hier `allowednet` nennen, zu erlauben, füge in diese Datei eine neue Sektion ein:

```
config subnet 'allowednet'
      option ipaddr '1.2.3.4'
      option netmask '255.255.255.255'
```

Alternativ geht das auch mit **uci**:

```
uci set luci_splash.allowednet=subnet
uci set luci_splash.allowednet.ipaddr=1.2.3.4
uci set luci_splash.allowednet.netmask=255.255.255.255
```

In beiden Fällen muss anschliessend der Splash mit **/etc/init.d/luci_splash restart** neu gestartet werden, damit die Änderungen wirksam werden.

5.9.7 Splash-Seite individualisieren

Es ist möglich, die Splash-Seite, die Benutzern angezeigt wird nach eigenen Vorstellungen anzupassen. So können z.B. ein prominenterer Hinweis auf den Betreiber oder Sponsor des Freifunkknotens eingefügt oder lokalitätsbezogene Informationen angezeigt werden.

Es kann entweder die komplette Splash-Seite ersetzt werden oder nur eigener Text in die Standardseite eingefügt werden.

Die Texte sollten in gültigem HTML geschrieben sein. Es können einige Marker verwendet werden, die bei der Ausgabe ersetzt werden:

Marker	Beschreibung
###COMMUNITY###	Name der Community
###COMMUNITY_URL###	URL zur Webseite der Community
###CONTACTURL###	URL zur lokalen Seite mit Kontaktinformationen
###LEASETIME###	Freigabezeit
###LIMIT###	Hinweis auf UP- und Downloadlimitierung
###ACCEPT###	Einbinden der *Akzeptieren* und *Ablehnen* Buttons

Splash Seite anpassen in LuCI

Die Splash-Seite kann im LuCI-Webinterface unter *Administration* → *Dienste* → *Client-Splash* → *Splash-Text* angepasst werden.

Soll der **komplette Text des Splash** angepasst werden dann gib ihn im oberen Textfeld *Bearbeiten des kompletten Splash-Textes* ein. Es ist wichtig, den Marker ###ACCEPT### einzufügen damit die Buttons angezeigt werden können.

Willst du **eigenen Text zum Splash hinzufügen** dann benutze das untere Textfeld *Einbinden von eigenem Text in die Default-Splashseite*.

Splash Seite anpassen auf der Shell

Um den **kompletten Text des Splash** zu ersetzen bearbeite die Datei `/usr/lib/luci-splash/splashtext.html` und füge dort deinen eigenen Text (gültiges HTML) ein.

Um nur **eigenen Text zusätzlich zum Standardtext** des Splashs anzuzeigen bearbeite `/usr/lib/luci-splash/splashtextinclude.html` und füge dort deinen eigenen HTML-Text ein.

5.9.8 Splash dauerhaft deaktivieren

Auf der von Meshkit installierten Firmware wird Splash üblicherweise automatisch installiert und eingerichtet. Will man den Splash dauerhaft deaktivieren geht das mit:

```
/etc/init.d/luci-splash stop
/etc/init.d/luci-splash disable
```

5.10 OLSR einrichten und konfigurieren

OLSR ist der Routingdämon der dafür sorgt, das Pakete im Mesh immer den richtigen Weg nehmen.

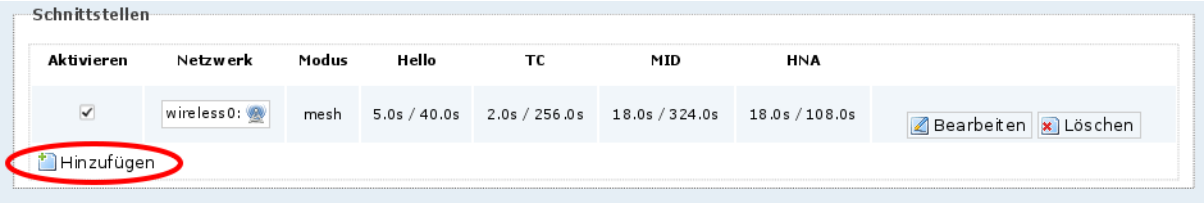
5.10.1 Ein Interface für OLSR konfigurieren

Hier wird gezeigt, wie man OLSR für eine Netzwerkschnittstelle in der Meshkit Firmware aktiviert.

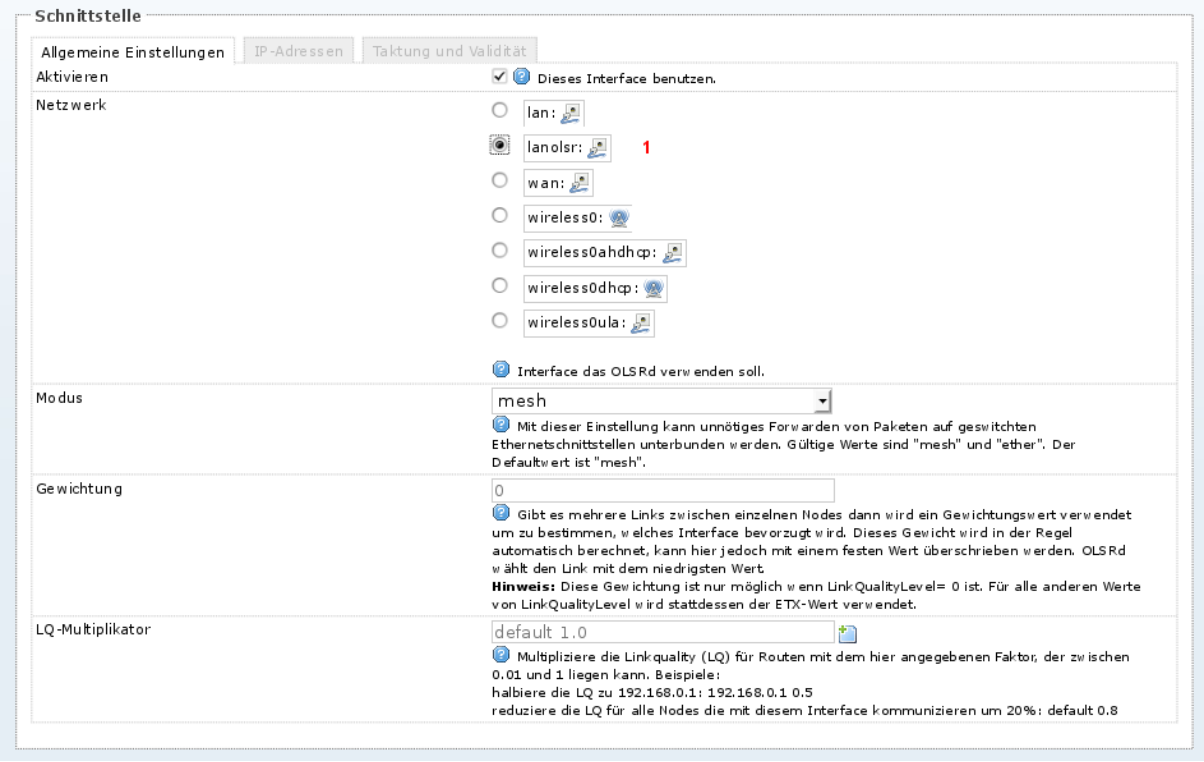
> **Warnung:** Hier wird nur gezeigt wie man OLSR für eine Schnittstelle aktiviert. Die Schnittstelle selbst muss auch konfiguriert werden und der `Freifunk`-Firewallzone hinzugefügt werden.

OLSR für eine Schnittstelle mit LuCI einrichten

Gehe zu *Administration* → *Dienste* → *OLSR*. Ganz unten siehst du die *Schnittstellen* Konfiguration:



Klicke dort auf *Hinzufügen*. Es öffnet sich eine neue Seite, um das hinzugefügte Interface zu konfigurieren:



Wähle dort bei *Netzwerk* das Netzwerk aus, für das *OLSR* konfiguriert werden soll. In diesem Beispiel wählen wir `lanolsr`, das in *Rechner der selbst OLSR benutzt* angelegt wurde.

Alle anderen Einstellungen können in der Regel auf ihren Defaultwerten gelassen werden. Speichere die konfiguration und starte olsrd neu durch einen Klick auf *Speichern & Anwenden*.

OLSR für eine Schnittstelle auf der Shell einrichten

Um OLSR für ein Netzwerk auf der Shell zu aktivieren fügt man am Ende von `/etc/config/olsrd` folgendes ein:

```
config Interface 'lanolsr'
     option ignore '0'
     option interface 'lanolsr'
     option Mode 'mesh'
```

Natürlich geht auch dies wieder mit **uci**:

```
uci set olsrd.lanolsr=Interface
uci set olsrd.lanolsr.ignore=0
uci set olsrd.lanolsr.interface='lanolsr'
```

```
uci set olsrd.lanolsr.Mode='mesh'
uci commit olsrd
```

Bemerkung: Ersetze `lanolsr` im den Beispielen oben durch den richtigen Namen des Netzwerks. Dies ist **nicht** der Name der phsyikalischen Schnittstelle sondern der Name unter dem OpenWrt dieses Netzwerk kennt, z.B. `lan`.

Anschliessend muss olsrd neu gestartet werden:

```
/etc/init.d/olsrd restart
```

5.11 OLSR Statusinformationen anzeigen

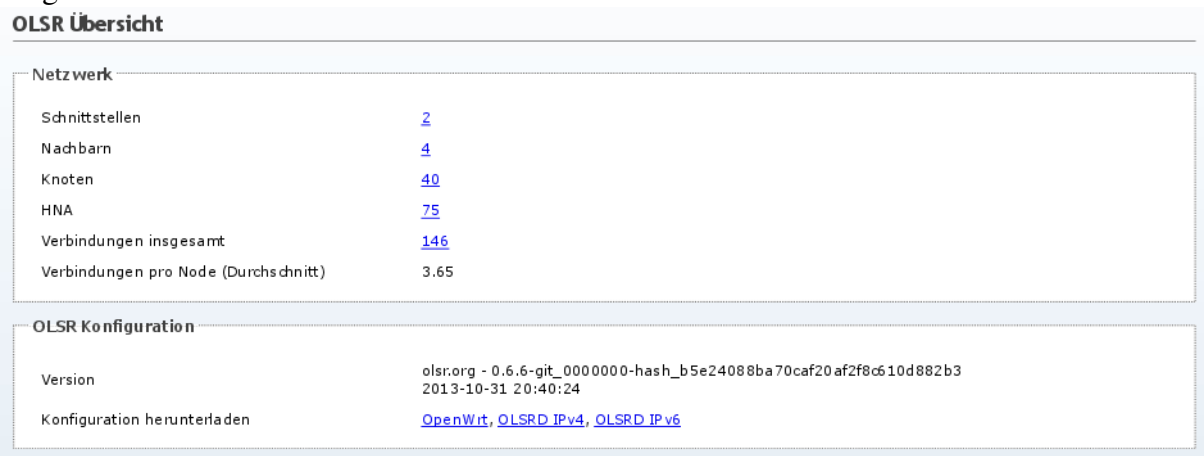
OLSR ist der Routingdämon der dafür sorgt, das Pakete im Mesh immer den richtigen Weg nehmen. Status zu OLSR kann sowohl im LuCI Webinterface als auch auf der Shell angezeigt werden

5.11.1 OLSR-Statusinformationen in LuCI

Statusinfos zu OLSR können sowohl im Administrationsmenu von LuCI (*Administration* → *Status* → *OLSR*) als auch im öffentlichen Teil des Webinterfaces (*Freifunk* → *OLSR*) angezeigt werden. Es werden jeweils die selben Informationen angezeigt.

OLSR-Übersicht

Zeigt eine Übersicht über den OLSR-Status.



In der oberen Sektion *Netzwerk* sind einige allgemeine Metriken zu OLSR zu sehen. Durch einen Klick auf die jeweilige Zahl kommt man zur entsprechenden Detailseite.

Metrik	Beschreibung
Schniistellen	Anzahl der Schnittstellen die OLSR nutzt
Nachbarn	Anzahl der direkten Nachbarn
Knoten	Anzahl aller bekannten Knoten im Mesh
HNA	Anzahl bekannter *HNA* im Mesh
Verbindungen insgesamt	Anzahl aller Links im Mesh
Verbindungen pro Node	durchschnittliche Verbindungen pro Knoten

In der unteren Sektion *OLSR-Konfiguration* sind Infos zur OLSR-Version sowie zur OLSR-Konfiguration zu finden. Es ist möglich, die auf dem Knoten laufende Konfiguration in verschiedenen Formaten herunterzuladen:

- *OpenWrt* - Konfiguration für OpenWrt (*UCI*)
- *OLSRD IPv4* - IPv4 Konfiguration für OLSRD
- *OLSRD IPv6* - IPv6 Konfiguration für OLSRD

5.11.2 OLSR-Statusinformationen auf der Shell

neigh.sh

neigh.sh ist ein kleines Script das Informationen zu den direkten Nachbarn ausgibt, z.B.:

```
root@freifunk:~# neigh.sh
Local      Remote     vTime LQ       NLQ      Cost
10.11.0.18 10.11.0.17 38004 0.894000 1.000000 1145
10.11.0.18 10.11.0.8  37751 1.000000 0.894000 1145

Local               Remote              vTime LQ       NLQ      Cost
fdca:ffee:ffa:12::1 fdca:ffee:ffa:11::1 39178 1.000000 1.000000 1024
fdca:ffee:ffa:12::1 fdca:ffee:ffa:8::1  36777 1.000000 1.000000 1024
```

jsoninfo plugin direkt befragen

Auf allen Knoten läuft `olsrd-jsoninfo`, das Informationen zu *OLSR* im *JSON*-Format ausgibt. Um Informationen vom jsoninfo-Plugin zu bekommen kann **nc** verwendet werden, z.B. um alle Informationen von jsoninfo zu bekommen:

```
echo "/all" | nc 127.0.0.1 9090
```

Ist man nur an bestimmten Informationen können auch nur diese abgefragt werden, z.B. um Informationen zu den Nachbarn zu bekommen:

```
echo "/neighbors" | nc 127.0.0.1 9090
```

Will man diese Informationen für den IPv6-OLSR erhalten dann ersetzt man `127.0.0.1` mit `::1`.

Folgende Detailinformationen können abgefragt werden:

Schlüssel	Beschreibung
all	Alle Informationen
neighbors	Nachbarn (inklusive 2-hop Nachbarn)
links	bestehende Links zu direkten Nachbarn
routes	Alle im netzwerk bekannten Routen
hna	Alle im Netzwerk bekannten *HNA*
mid	Alle im Netzwerk bekannten *MID*
gateways	Smart-Gateways im Netz (nur wenn das Smart Gateway Plugin aktiv ist)
interfaces	Informationen zu Interfaces auf denen OLSR läuft
status	Gibt Infos zu neighbors, links. routes, hna, mid, gateways und interfaces kombiniert aus
config	aktuelle Konfiguration
olsrd.conf	aktuelle Konfiguration im Format des `olsrd.conf` Konfigurationsfiles
plugins	Informationen zu geladenen Plugins

5.12 Firewall

Standardmässig ist auf jedem Node eine Firewall aktiv. Die Regeln der Firewall bestimmen, welche eigehenden Datenpakete das Node erreichen bzw- weitergeleitet werden dürfen.

Hinweis: Ausführlichere Doku zur OpenWrt Firewall und einzelnen Optionen gibt es im OpenWrt Wiki: http://wiki.openwrt.org/doc/uci/firewall (englisch)

5.12.1 Firewallzonen

OpenWrt benutzt ein Konzept mit Zonen, denen eine oder mehrere Schnittstellen zugeordnet werden. Standardmässig gibt es in der Meshkit Firmware drei Zonen:

Zone	Beschreibung	INPUT	FORWARD	NAT
WAN	Zone für Interfaces die direkt mit dem Internet verbunden sind	verboten	verboten	ja
LAN	Zone für lokale Interfaces	erlaubt	erlaubt in alle anderen Zonen	nein
Freifunk	Zone für Interfaces die Teil des öffentlichen Meshs sind	verboten, einzelne Ports erlaubt	erlaubt nach WAN und Freifunk	teilweise

Das heisst:

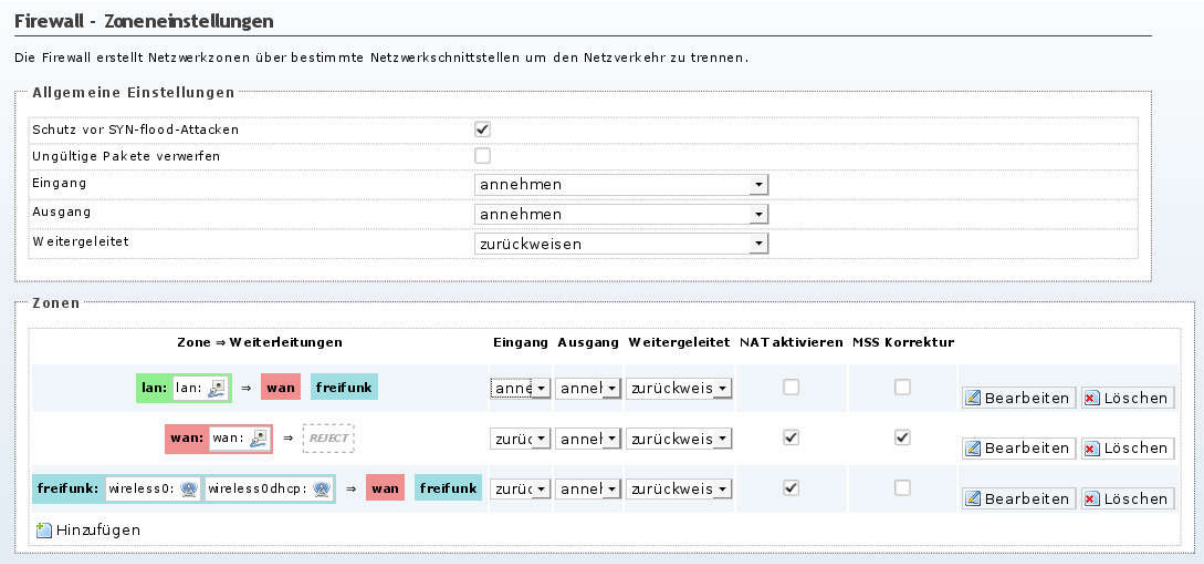
- die WAN-Zone erlaubt per Default weder einkommende Verbindungen (INPUT) noch wird Traffic der über die WAN-Zone hereinkommt in andere Zonen weitergeleitet (FORWARD). Ausgehender Datenverkehr wird genatted, d.h. die Quelladresse auf die Adresse der ausgehenden Schnittstelle umgeschrieben (Masquerading).

- Die LAN-Zone erlaubt jeglichen einkommenden Verkehr sowie die Weiterleitung von Paketen in beliebige Zonen
- Die Freifunk-Zone erlaubt keinen eingehenden Verkehr, jedoch werden einige Ports die für den betrieb des Knotens sind einzeln erlaubt (SSH, Weboberfläche, *DHCP*, OLSR-Pakete usw.). Traffic darf in die Zonen WAN und LAN. jedoch nicht in die Zone LAN weitergeleitet werden. Weitergeleiteter Verkehr wird in der Regeln nicht genatted. Hier gibt es jedoch eine Ausnahme: Wird für *DHCP*-Clients ein IP-Netzwerk benutzt, das vom lokalen Node nicht als *HNA* angekündigt wird, dann werden Pakete aus diesem *DHCP*-Netzwerk genatted.

Zoneneinstellungen im Webinterface

Die Zoneneinstellungen der Firewall findet man im LuCI-Webinterface unter *Administration → Netzwerk → Firewall → Allgemeine Einstellungen*

> **Warnung:** Hier solltest du Einstellungen nur verändern, wenn du genau weisst was du tust.



Zoneneinstellungen auf der Shell

Alle Firewalleinstellungen und damit auch die Einstellungen für Zonen sind in `/etc/config/firewall` gespeichert. Diese Datei kann direkt bearbeitet werden. Folgender Ausschnitt zeigt die Einstellungen für die **WAN-Zone**:

```
config zone
        option name 'wan'          # Die Zone heisst wan
        list network 'wan'         # Die Netzwerke wan und wan6
                                   # gehören der Zone an
        list network 'wan6'        # Netzwerke werden definiert in
                                   # :file:'/etc/config/network'
        option input 'REJECT'      # Eingehenden Verkehr ablehnen
        option output 'ACCEPT'     # Ausgehenden Verkehr akzeptieren
```

```
        option forward 'REJECT'    # Eingehenden Verkehr nicht
                                   # weiterleiten
        option masq '1'            # NAT (Masquerading) für weiter-
                                   # geleiteten ausgehenden Verkehr
        option mtu_fix '1'         # MSS clamping Regeln für
                                   # ausgehenden Verkehr
                                   # (gegen MTU-Probleme)
        option local_restrict '1'  # siehe Bemerkung unten
```

Bemerkung: Die Option `local_restrict` ist keine Option die OpenWrt von Haus aus kennt. Sie ist freifunkspezifisch und sorgt dafür, dass Regeln eingefügt werden, die den Zugriff aufs Heimnetzwerk verbieten wenn der Freifunkrouter mit seiner WAN-Buchse am eigenen LAN-Netzwerk angeschlossen ist, siehe: *Der Standardfall: Freifunk Router am Heimnetzwerk*.

Eine Konfiguration über **uci** ist zwar prinzipiell möglich, jedoch umständlich da die Sektionen in der Konfigurationsdatei keine Namen haben.

Nach Änderungen an der Firewall Konfiguration ist es notwendig die Firewall durch das Kommando **fw3 reload** neu zu starten.

5.12.2 Port Forwarding

Port Forwarding wird eingesetzt, um einzelne Ports auf Rechnern hinter dem Router zu öffnen. Dies ist z.B notwendig, wenn Ports auf einem Rechner im LAN-Netzwerk aus dem Internet erreichbar sein sollen.

Das Ganze erklärt sich am besten durch ein Beispiel:

Das Setup ist wie in *Freifunk Router direkt am Internetzugang* beschrieben. Der Freifunkrouter ist mit seiner WAN-Buchse direkt am Internet angeschlossen. An einer der LAN-Buchsen ist ein Rechner mit der IP 192.168.1.249 angeschlossen, auf dem ein Webserver auf Port 80 läuft. Dieser Webserver soll nun auch aus dem Internet erreichbar sein, und zwar unter Port 8080.

Das Portforwarding kann nun entweder über LuCI oder die Shell eingerichtet werden.

Portforwarding mit LuCI einrichten

Gehe zu *Administration* → *Netzwerk* → *Firewall* → *Portweiterleitungen*. Dort kannst du eine neue Portweiterleitung einrichten:

Gib die Daten wie oben gezeigt ein. Erklärung der einzelnen Optionen:

Option	Beschreibung	Default
Name	Ein Name für diese Port Forwarding Regel, hier web	keiner
Protokoll	Nur Pakete von diesem Protokolltyp forwarden, hier TCP+UDP Es hätte auch TCP alleine gereicht, da HTTP nur TCP verwendet.	TCP+UDP
Externe Zone	Firewallzone auf der den Router die Anfrage erreicht, hier `wan`	
Externer Port	Dieser Port soll weitergeleitet werden, hier `8080`	
Interne Zone	Die Firewallzone in der sich der Zielrechner der Weiterleitung befindet, hier `lan`	
Interner Port	An diesen Internen Port des Zielrechners werden Pakete weitergeleitet, hier `80`	wie Externer Port

Nachdem alle Optionen ausgefüllt wurden klicke auf den *Speichern & Anwenden* Button. Die Weiterleitung wird jetzt gespeichert, die Firewall neu geladen damit die Weiterleitung aktiv wird und die Seite des Webinterfaces neu geladen.

Die Seite sieht mit der neuen Portweiterleitung nun so aus:

Es gibt hier nun auch die Möglichkeit, diese Regel zu (De-)aktivieren sowie bei mehreren Weiterleitungsregeln die Reihenfolge der Regeln zu ändern. Mit einem Klick auf *Bearbeiten* kann die Regel bearbeitet, mit einem Klick auf *Löschen* gelöscht werden.

Portweiterleitung auf der Shell

Die Weiterleitung lässt sich auch auf der Shell anlegen. Entweder man editiert `/etc/config/firewall` direkt und fügt folgende Sektion hinzu:

```
config redirect web
        option target 'DNAT'
        option src 'wan'
        option dest 'lan'
        option proto 'tcp udp'
        option src_dport '8080'
        option dest_ip '192.168.1.249'
        option dest_port '80'
        option name 'web'
```

oder direkt über **uci** Kommandos:

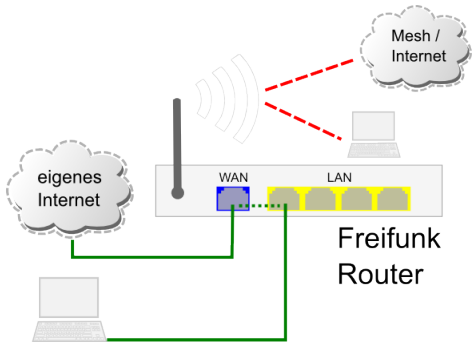
```
uci set firewall.web=redirect
uci set firewall.web.target=DNAT
uci set firewall.web.src=wan
uci set firewall.web.dest=lan
uci set firewall.web.proto='tcp udp'
uci set firewall.web.src_dport=8080
uci set firewall.web.dest_ip=192.168.1.249
uci set firewall.web.dest_port=80
uci set firewall.web.name=web
uci commit firewall
```

In beiden Fällen muss die Konfiguration der Firewall durch **fw3 reload** neu eingelesen werden.

5.13 Policy Routing

Policy Routing ist nicht ganz einfach und im Idealfall musst du als Benutzer darüber auch nicht viel wissen, da es einfach im Hintergrund dafür sorgt, dass das Mesh-netzwerk richtig funktioniert. Dennoch sollen hier einige Grundlagen des Policy Routings in der Meshkit Firmware erklärt werden, damit die Möglichkeit besteht zu begreifen was hier passiert.

5.13.1 Was ist Policy Routing und wofür ist es gut?



Das `freifunk-policyrouting`-Paket ist standardmässig installiert und wird benötigt, um Traffic je nach Herkunft unterschiedlich weiterleiten zu können.

Eigener Traffic soll über die eigene Internetleitung geschickt werden, während Traffic aus dem Freifunknetz auch nur ins Freifunknetz weitergeleitet werden darf. Als eigener Traffic ist per Default aller Traffic definiert, der entweder **lokal**, also vom Freifunkrouter selbst generiert wurde oder der aus der Firewall-Zone `LAN` kommt. Als **Freifunktraffic** wird alles betrachtet, was das Node über ein Interface erreicht hat, das zur Zone `Freifunk` gehört. Für eine Erklärung der Zonen siehe: *LAN, WAN und Freifunk - Die einzelnen Zonen im Router*.

Das Bild verdeutlicht dies:

- Datenverkehr von dem am `LAN` angeschlossenen PC wird über die eigene Leitung verschickt, lokal vom Knoten selbst ausgehender Traffic ebenfalls

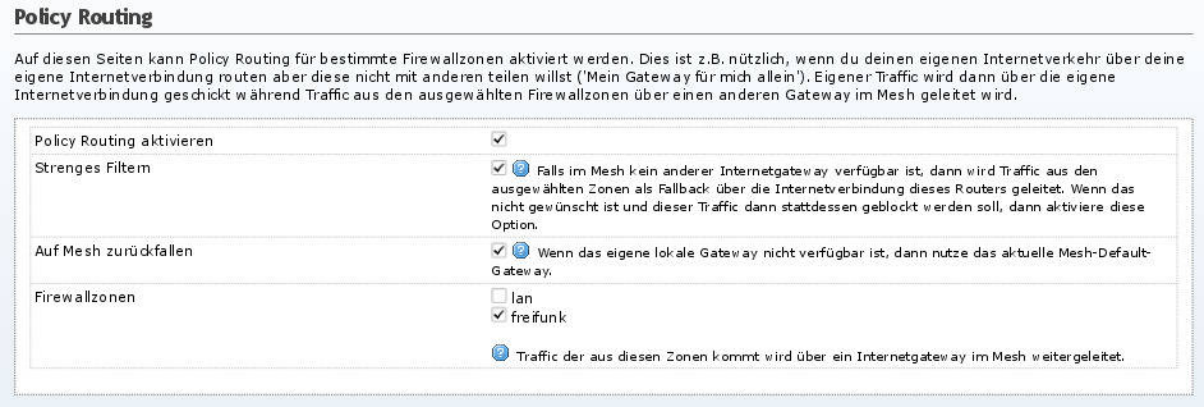
- Verkehr der über `Freifunk` hereinkommt (in diesem Fall über WLAN) darf das Node nur Richtung Freifunk verlassen. In diesem Fall wird dieser Verkehr ebenfalls wieder über WLAN ins Mesh weitergeschickt, um darüber sein Ziel zu erreichen.

5.13.2 Konfiguration von Policy Routing

> **Warnung:** Diese Einstellungen sind komplex und es ist leicht Dinge kaputt zu machen. Ändere an den EInstellungen also nur etwas, wenn du genau weisst was du tust.

Mit Luci

Policy Routing kann in LuCI unter *Administration* → *Freifunk* → *Policy Routing* konfiguriert werden. Das Paket das diese Funktion für LuCI mitbringt ist `luci-app-freifunk-policyrouting` und ist in der Regel schon installiert. Ansonsten muss es noch nachinstalliert werden, siehe *Pakete installieren*.



> **Hinweis:** Will man auch Traffic aus der `LAN`-Zone nur übers Freifunknetz schicken dann fügt wählt man bei *Firewallzonen* zusätzlich auch noch `lan` aus.

Auf der Shell

Die Konfiguration für das policy Routing befindet sich in `/etc/config/freifunk-policyrouting` und kann entweder direkt bearbeitet werden oder mit **uci** verändert werden.

```
config 'settings' 'pr'
     option 'strict' '1'
     option 'fallback' '1'
     option 'zones' 'freifunk'
     option 'enable' '1'
```

Zum Bearbeiten mit **uci**:

```
uci set freifunk-policyrouting.pr.strict=1
uci set freifunk-policyrouting.pr.fallback=1
uci set freifunk-policyrouting.pr.enable=1
uci set freifunk-policyrouting.pr.zones=freifunk
uci commit freifunk-policyrouting.pr.zones
```

Beschreibung der einzelnen Optionen:

Option	Beschreibung	Default
strict	Wenn keine Route ins Internet über das Freifunknetz verfügbar ist dann würde Traffic aus `Freifunk` über die eigene Leitung geroutet. Der Wert 1 verhindert dies durch Einfügen einer Firewallregel die dies verbietet. Will man in diesem Fall erlauben den Traffic über das eigene Internet zu routen dann den Wert auf 0 setzen	1
fallback	Gibt an, ob bei Ausfall der eigenen Internetleitung der eigene Traffic über Freifunk ins Internet geroutet werden soll. Wird als Wert 1 gesetzt dann darf Freifunk als Fallback verwendet werden.	0
enable	1 aktiviert Policy Routing, 0 deaktiviert es	0
zones	Zonen die für die die Policy Routing Regeln gelten sollen.	

Wurden alle Einstellungen gemacht/verändert dann muss Policy Routing noch mit

```
/etc/init.d/freifunk-policyrouting restart
```

neu gestartet werden, damit diese Konfiguration aktiv wird.

Hinweis: Will man auch Traffic aus der `LAN`-Zone nur übers Freifunknetz schicken dann fügt man der Option `zones` noch `lan` hinzu, also zones="freifunk lan".

5.13.3 Infos zu Routingtabellen und zum Debuggen

Zum Anzeigen und Verändern der Routingtabellen und -regeln wird **ip** verwendet. Wichtige Kommandos von ip sind in diesem Zusammenhang:

Kommando	Aktion
ip rule show	zeigt alle Regeln
ip route show	zeigt alle Routen aus der `main` Routingtabelle
ip route show table olsr	zeigt alle Routen aus der `olsr` Routingtabelle

Zunächst wollen wir uns für einen Überblick alle Regeln ansehen und geben dazu auf der Shell des Routers **ip rule show** ein. Als Ausgabe erscheint auf einem Knoten auf dem Policyrouting für die Freifunkzone aktiviert wurde:

```
0:          from all lookup local
1000:       from all lookup olsr
2000:       from all lookup localnets
20000:      from all iif wlan0 lookup olsr-default
20000:      from all iif wlan0-1 lookup olsr-default
20001:      from all iif wlan0 unreachable
20001:      from all iif wlan0-1 unreachable
32766:      from all lookup main
32767:      from all lookup default
100000:     from all lookup olsr-default
```

Die Zahl in der ersten Spalte gibt die **Priorität** an. In dieser Reihenfolge werden die Regeln durchlaufen. Danach folgen **Bedingungen** die zutreffen müssen, damit ein Paket das weitergeleitet werden soll diese Regel benutzt. Zum Beispiel meint `from all iif wlan0-1` von allen IPs, von denen Pakete über die Schnittstelle `wlan0-1` empfangen wurden. Schliesslich folgt eine Anweisung, wie mit dem Paket zu verfahren ist. `unreachable` besagt, wir brechen hier ab wenn zuvor keine passende Route gefunden wurde. `lookup` dagegen ist eine Anweisung, zu welcher Routingtabelle gesprungen werden soll um dort eine passende Route zu finden. Wird keine passende Route in dieser Tabelle gefunden dann wird mit der nächsten regel fortgefahren.

Für Policy Routing sind folgende Routingtabellen von Bedeutung:

Tabelle	Beschreibung
local	enthält lokale Hostrouten
olsr	enthält alle von *OLSR* empfangenen Routen mit Ausnahme der Defaultroute
olsr-default	enthält die von *OLSR* empfangene Defaultroute (sofern vorhanden)
localnets	Enthält Routen zu direkt erreichbaren Netzwerken
main	Dies ist die Hauptroutingtabelle von Linux. Hier steht unter anderem die eigene Defaultroute über den eigenen Internetzugang (sofern vorhanden).
default	wird nicht verwendet

Zwei Beispiele sollen helfen zu verstehen, wie Traffic aus verschiedenen Zonen anders behandelt wird:

5.13.4 Fluss von Traffic aus der LAN-Zone oder vom Router selbst

Von einem an `LAN` angeschlossenen Rechner oder vom Node selbst soll eine Verbindung mit einer Adresse im Internet aufgebaut werden. Daher muss eine Defaultroute gefunden werden.

Zunächst werden die Regeln 0, 1000 und 2000 durchlaufen und dementsprechend die Tabellen `local`, `olsr` und `localnets` befragt. Hier wird jedoch keine Defaultroute gefunden. Die Regeln mit den Prioritäten 20000 und 20001 werden übersprungen, da die Bedingungen `iif wlan0` bzw. `iif wlan0-1` nicht zutreffen. Die Regel mit der Priorität 32766 verweist nun auf die `main` Routingtabelle. Wird hier nun eine Defaultroute gefunden, dann benutzt das Paket die gefundene Route. Wird keine gefunden suchen wir weiter. Die `default` Tabelle ist

leer und wird daher übersprungen. Wurde ausgewählt dass Fallback übers Freifunknetz möglich sein soll erreichen wir schliesslich die Regel mit der Priorität 100000, die die Anweisung enthält, in der Tabelle `olsr-default` nachzuschauen. Wird dort eine defaultroute gefunden wird diese benutzt und das Paket übers Mesh geschickt. Wenn nicht dann hatten wur kein Glück. In diesem Fall kennt das Node keine Route ins Internet.

5.13.5 Fluss von Traffic aus der Freifunk-Zone

Ein über WLAN verbundener Rechner oder Access Point will eine Verbindung mit einer Adresse im Internet aufbauen. Der Rechner ist über die Schnittstelle `wlan0` verbunden, die zur Zone `Freifunk` gehört. Es muss nun wiederum eine Defaultroute gefunden werden.

Zunächst werden die Regeln 0, 1000 und 2000 durchlaufen und dementsprechend die Tabellen `local`, `olsr` und `localnets` befragt. Hier wird jedoch keine Defaultroute gefunden. Eine der Regeln mit der Priorität 20000 trifft zu, es wird also die Tabelle `olsr-default` befragt. Wird in dieser Tabelle eine Defaultroute gefunden, dann wird diese benutzt. Wird keine gefunden geht es weiter mit den Regeln mit Priorität 20001 die hier sind, weil die Option `strict` gewählt wurde. Die Suche nach einer Route wird hier durch die Anweisung `unreachable` abgebrochen und dem Rechner mitgeteilt, dass der Knoten keine defaultroute für ihn kennt. Hätte man nicht die Option `strict` gewählt dann würde als nächstes die Tabelle `main` befragt und dort die eigene Defaultroute des knotens gefunden und benutzt.

5.13.6 Links

- Policy Routing im Freifunk Wiki
- Quellcode von freifunk-policyrouting

5.14 Dateien vom/zum Router kopieren

Setzt voraus: *Mit dem Router verbinden*

Will man Dateien vom/zum Router kopieren, dann ist dies mit SCP möglich, das Dateien über SSH bzw. das SCP Protokoll kopiert.

5.14.1 Von Linux aus

Mit scp

Das **scp**-Kommand ist auf den meisten Distributionen bereits installiert.

Um eine **Datei zum Router zu kopieren** gibt man in der Kommandozeile des Linux-Systems folgendes ein:

```
scp <Datei> root@<ip-des-routers>:/tmp
```

Dadurch wird die Datei nun, nachdem man sofern benötigt noch das Passwort eingegeben hat, in den `/tmp`-ordner des Routers kopiert. Will man ganze Verzeichnisse rekursiv kopieren dann benutzt man für scp den "-r" Parameter, also z.B.

```
scp -r <Verzeichnis> root@<ip-des-routers>:/tmp
```

Um eine Datei vom Router auf den eigenen Linux-Rechner zu kopieren werden einfach die Parameter umgedreht. Um zum Beispiel die Datei `/etc/config/wireless` zu sichern kann folgendes KOmmando verwendet werden:

```
scp root@<ip-des-routers>:/etc/config/wireless .
```

Dies kopiert die Datei in den aktuellen Ordner. Sollen ganze Verzeichnisse gesichert werden dann ebenfalls wieder den "-r"-Parameter verwenden.

Hinweis: Diese Methode kann auch verwendet werden, um Dateien von einem Freifunkrouter zu einem anderen zu kopieren.

Mit fish

In vielen Dateimanagern kann auf entfernte Dateisysteme zugegriffen werden, indem man in der Adresszeile die Adresse **fish://<ip-des-routers>** eingibt.

5.14.2 Von Windows aus

Unter Windows kann WinSCP verwendet werden, um Dateien vom/zum Freifunkrouter zu kopieren.

KAPITEL 6

Fortgeschrittene Konfiguration

Da die Meshkit Firmware auf dem vielseitigen OpenWrt basiert, gibt es viele Möglichkeiten die Software auf dem Router für verschiedenste Einsatzzwecke anzupassen.

6.1 Angeschlossene Computer im Freifunknetz erreichbar machen

Sind eigene Computer über die LAN oder die WAN-Schnittstelle mit dem Router verbunden (siehe *Einbinden des Freifunkrouters ins eigene Netzwerk*), dann sind diese aus dem Freifunknetz heraus nicht direkt erreichbar. Möchte man auf diesen nun Dienste (siehe: *Eigene Dienste anbieten* anbieten, auf die aus dem Freifunknetz zugegriffen werden kann, dann muss der Freifunkrouter entsprechend umkonfiguriert werden.

6.1.1 Einzelne Dienste mit Portforwarding verfügbar machen

Das ist der einfachste Weg, der keine Eingriffe ins Netzwerk an sich erfordert. Es wird eine (oder auch mehrere) Regel für Portforwarding erstellt. Portforwarding funktioniert dabei so, dass Zugriffe auf den Port einer externen Adresse auf einen Rechner hinter dem Freifunkrouter weitergeleitet werden.

Zum Einrichten von Portforwards siehe *Port Forwarding*. Als `Externe Zone` ist in diesem Fall `Freifunk` zu wählen, schliesslich sollen ja Pakete die an einen Port der Freifunk- Schnittstelle(n) gerichtet sind weitergeleitet werden.

6.1.2 Rechner oder Netzwerke mit HNA ankündigen

Mit *HNA* (Host Network Announcement) kann auf dem Node ein verfügbarer IP-Bereich oder auch nur eine einzelne Adresse im Netzwerk bekannt gemacht werden. Dadurch wird ein angeschlossener Rechner direkt im Mesh erreichbar, muss jedoch selbst nicht *OLSR* nutzen.

Um einen Rechner im WAN oder LAN (siehe *Einbinden des Freifunkrouters ins eigene Netzwerk*) per *HNA* erreichbar zu machen, benötigt man ein eigenes kleines Netzwerk aus dem

Bereich der eigenen Freifunk Community und sollte dieses Netzwerk (bzw. darin liegende einzelne IPs registrieren, siehe *IP-Adresse(n) registrieren*. Soll nur ein einzelner Rechner per HNA angekündigt werden, dann reicht es ein Netzwerk mit der *Netzmaske* /30 zu registrieren. Sollen es mehr Rechner werden dann dieses Netz entsprechend grösser wählen (z.B. /29 für 6 Rechner oder /28 für 14 Rechner).

Hinweis: Um Rechner über HNA erreichbar zu machen müssen die IP-Adressen auf dem Freifunkrouter **und** auf dem Rechner angepasst werden.

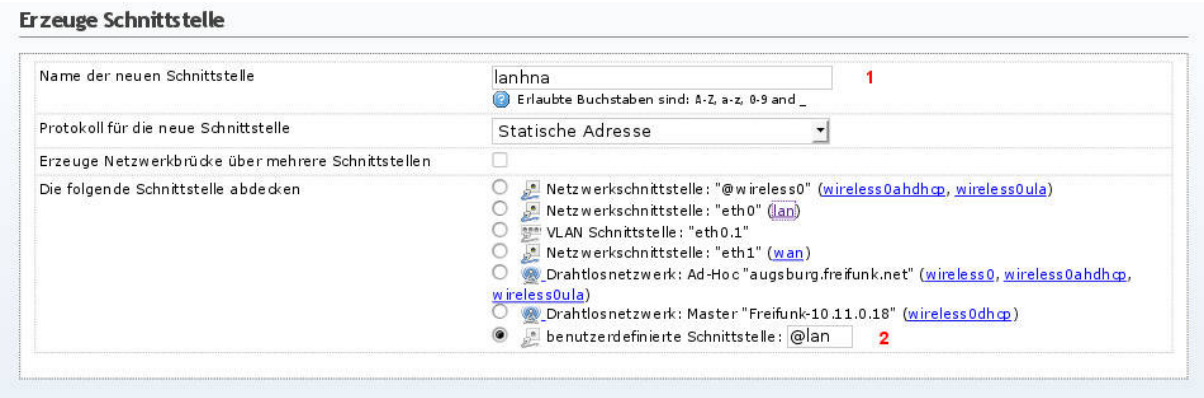
Beispiel

Im Augsburger Freifunknetz, das als Mesh-Netzwerk 10.11.0.0/18 benutzt soll ein am LAN des Freifunkrouters angeschlossener Rechner per HNA verfügbar gemacht werden. Dazu wird zunächst das Netzwerk 10.11.9.0/30, also die IPs 10.11.9.0 bis 10.11.9.3 reserviert. Das ergibt 2 nutzbare IPs:

- 10.11.9.1 soll als Alias-IP (also zusätzliche IP) für LAN auf dem Freifunkrouter verwendet werden
- 10.11.9.2 soll die IP-Adresse des angeschlossenen Rechners werden.

Einrichtung eines HNA-Netzwerks in LuCI

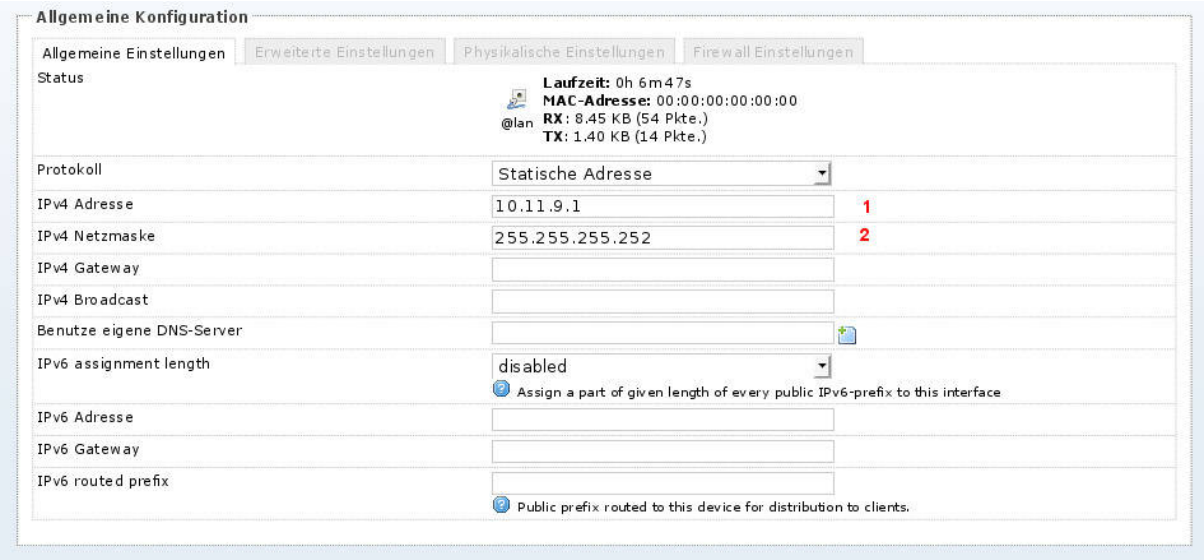
Gehe zu *Administration* → *Netzwerk* → *Schnittstellen* und klicke unten auf *Neue Schnittstelle hinzufügen...*. Es öffnet sich nun eine Seite zum konfigurieren der neuen Schnittstelle:



Trage dort bei *Name der neuen Schnittstelle* (1) einen Namen ein. Da wir hier ein HNA-Alias-Interface für LAN anlegen wollen verwenden wir als Namen `lanhna`. Wähle unten bei *Die folgende Schnittstelle abdecken Benutzerdefinierte Schnittstelle* aus und trage in das Feld hinten `@lan` ein, damit die neue Schnittstelle als Alias-Interface der `LAN`-Schnittstelle erzeugt wird.

Hinweis: Ist der Rechner statt an den `LAN`-Buchsen am `WAN` des Freifunkrouters angeschlossen gibst du hier `@wan` statt `@lan` ein.

Klicke abschliessend auf *Absenden* um zu einer weiteren Konfigurationsseite zu kommen:



Trage hier bei *IPv4-Adresse* (1) eine Adresse ein, in unserem Beispiel also 10.11.9.1. Bei *IPv4 Netzmaske* wird die *Netzmaske* des Netzwerks eingetragen. In unserem Beispiel wäre das `255.255.255.252`.

Wechsle danach zum Tab *Firewall Einstellungen*:

Wähle hier bei *Firewallzone anlegen / zuweisen* die `Freifunk` Zone aus und beende das Anlegen des Interfaces durch einen Klick auf *Speichern & Anwenden*.

Als nächstes muss noch ein *HNA*-Eintrag für *OLSR* angelegt werden. Gehe dafür zu *Administration* → *Dienste* → *OLSR* → *HNA-Ankündigungen* und klicke dort in der Sektion *Hna4* auf *Hinzufügen*. Trage hier bei *Netzwerkadresse* (1) die Startadresse des Alias-Netzwerks ein, in unserem Beispiel 10.11.9.0. Bei *Netzmaske* (2) trage die *Netzmaske* dieses Netzwerks ein, hier ist das 255.255.255.252.

Klicke anschliessend auf *Speichern & Anwenden*.

Einrichtung eines HNA-Netzwerks auf der Shell

1. Alias Interface anlegen

Zunächst muss ein *Alias-Interface* angelegt werden. Dazu wird am Ende von `/etc/config/network` folgendes eingefügt:

```
config interface 'lanhna'
option proto 'static'
option ifname '@lan'
option ipaddr '10.11.9.1'
option netmask '255.255.255.252'
```

Alternativ kann das auch mit **uci** erledigt werden:

```
uci set network.lanhna='interface'
uci set network.lanhna.proto='static'
uci set network.lanhna.ifname='@lan'
uci set network.lanhna.ipaddr='10.11.9.1'
uci set network.lanhna.netmask='255.255.255.252'
```

2. Alias Schnittstelle zur Firewallzone ‘‘freifunk‘ hinzufügen

In `/etc/config/firewall` muss die `freifunk`-Zone bearbeitet und dort das eben angelegte Alias-Interface `lanhna` für `option network` hinzugefügt werden.

```
config zone 'zone_freifunk'
     option name 'freifunk'
     option input 'REJECT'
     option forward 'REJECT'
     option output 'ACCEPT'
     option masq '1'
     list masq_src 'lan'
     list masq_src 'wireless0dhcp'
     list masq_src 'wireless0ahdhcp'
     option network 'wireless0 wireless0dhcp lanhna'
```

Auch das ist wiederum mit **uci** direkt möglich:

```
net="$(uci get firewall.zone_freifunk.network)"
uci set firewall.zone_freifunk.network="$network lanhna"
uci commit firewall
```

> **Warnung:** Bei uci müssen wir hier den Weg gehen, zunächst herauszufinden, welche Netze bereits zur Zone gehören (Die Variable `network` aus der ersten Zeile).

3. HNA Eintrag in der OLSR-Konfiguration erstellen

Um einen *HNA*-Eintrag zu Erstellen muss am Ende von `/etc/config/olsrd` folgendes eingefügt werden:

```
config Hna4 'lanhna'
     option netaddr '10.11.9.0'
     option netmask '255.255.255.252'
```

oder alternativ wieder mit **uci**:

```
uci set olsrd.lanhna='Hna4'
uci set olsrd.lanhna.netaddr='10.11.9.0'
uci olsrd.lanhna.netmask='255.255.255.252'
uci commit olsrd
```

4. Dienste neu starten

```
/etc/init.d/network restart
/etc/init.d/olsrd restart
```

Konfiguration des angeschlossenen Rechners

Der per *HNA* angeschlossene Rechner muss eine IP-Adresse aus dem Bereich verwenden, der per HNA angekündigt wird. Dies kann die primäre IP des Rechners, aber auch eine Alias-IP sein. In unserem Beispiel wäre die IP des Rechners 10.11.9.2 und die *Netzmaske* 255.255.255.252.

6.1.3 Rechner der selbst OLSR benutzt

Indem ein Rechner/Server selbst *OLSR* "spricht" kann er direkt aus dem Mesh erreicht werden.

Auf dem Freifunkrouter soll daher ein **Alias-Netz** eingerichtet werden, das auf `LAN` zusätzlich zur LAN-IP eine IP-Adresse aus dem IP-Bereich der jeweiligen Community hat. Der anzuschliessende Rechner braucht ebenfalls eine IP aus diesem IP-Bereich. Siehe *IP-Adresse(n) registrieren*. Zusätzlich muss auf beiden, Freifunkrouter und angeschlossener PC *OLSR* für die jeweilige Schnittstelle eingerichtet werden.

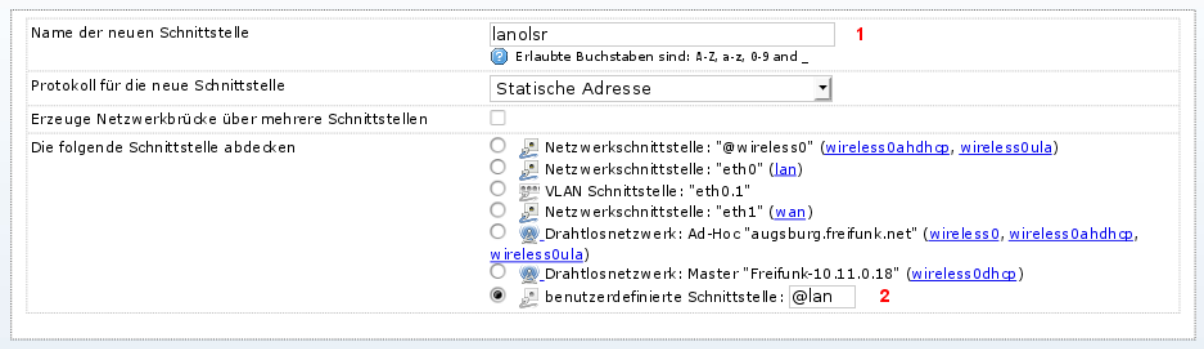
Hinweis: Wenn du schon vorher weisst dass entweder per `LAN` oder `WAN` ein oder mehrere Rechner per OLSR angeschlossen werden sollen, dann kannst du auch im Meshkit beim erstellen des Images direkt OLSR als Protokoll für dieses Netzwerk wählen. Das betreffende Netzwerk wird dann direkt für OLSR-Betrieb konfiguriert. **Es gibt dann kein Alias, d.h. die urprüngliche Funktionalität wird nicht erhalten.** Siehe: *Konfiguration von LAN im Meshkit* bzw. *Konfiguration von WAN im Meshkit*.

Beispiel für diese Konfiguration

Der Freifunkrouter soll die IP 10.11.9.100 bekommen, der Rechner die IP 10.11.9.101. Als *Netzmaske* wird /18, also 255.255.192.0 verwendet.

Einrichtung eine Alias-Schnittstelle für OLSR unter LuCI

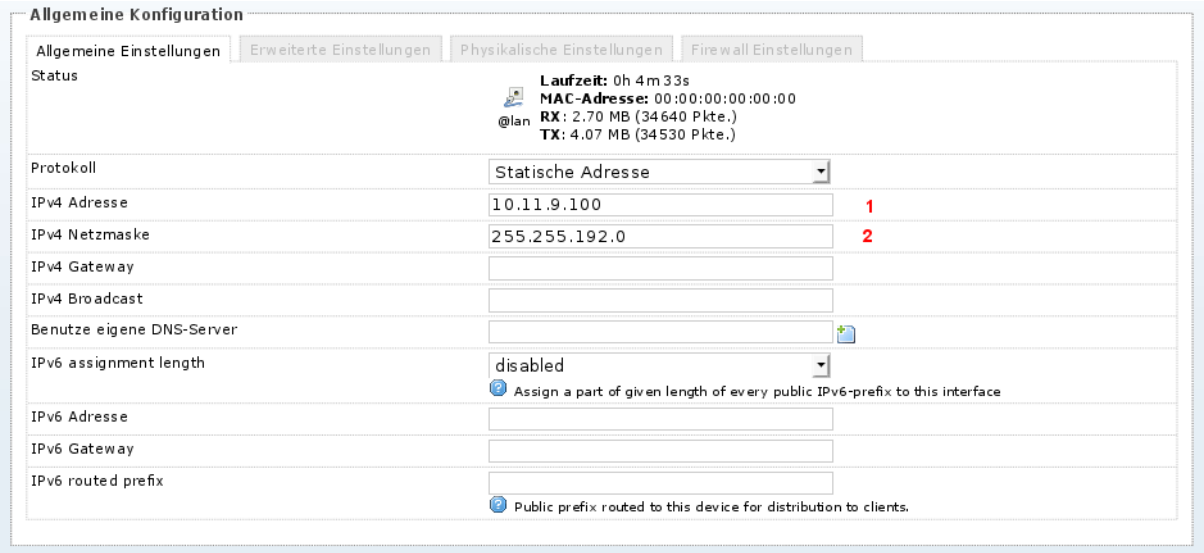
Gehe zu *Administration* → *Netzwerk* → *Schnittstellen* und klicke unten auf *Neue Schnittstelle hinzufügen...*. Es öffnet sich nun eine Seite zum konfigurieren der neuen Schnittstelle:



Trage dort bei *Name der neuen Schnittstelle* (1) einen Namen ein. Da wir hier ein OLSR-Alias-Interface für LAN anlegen wollen verwenden wir als Namen `lanolsr`. Wähle unten bei *Die folgende Schnittstelle abdecken Benutzerdefinierte Schnittstelle* aus und trage in das Feld hinten `@lan` ein, damit die neue Schnittstelle als Alias-Interface der `LAN`-Schnittstelle erzeugt wird.

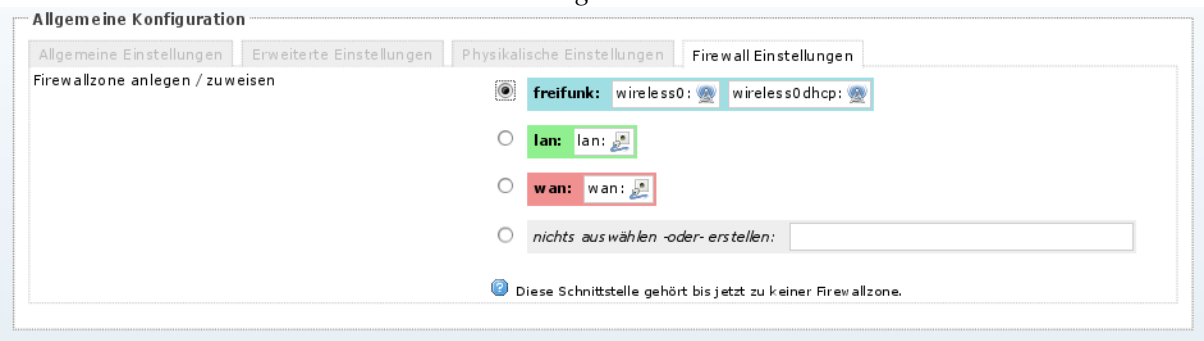
Hinweis: Ist der Rechner statt an den `LAN`-Buchsen am `WAN` des Freifunkrouters angeschlossen gibst du hier `@wan` statt `@lan` ein.

Klicke abschliessend auf *Absenden* um zu einer weiteren Konfigurationsseite zu kommen:



Trage hier bei *IPv4-Adresse* (1) eine Adresse ein, in unserem Beispiel also 10.11.9.100. Bei *IPv4 Netzmaske* wird die *Netzmaske* des Netzwerks eingetragen. In unserem Beispiel wäre das `255.255.192.0`.

Wechsle danach zum Tab *Firewall Einstellungen*:



Wähle hier bei *Firewallzone anlegen / zuweisen* die `Freifunk` Zone aus und beende das Anlegen des Interfaces durch einen Klick auf *Speichern & Anwenden*.

Jetzt muss noch *OLSR* für diese neu angelegte Schnittstelle aktiviert werden. Verwende dabei `lanolsr` als Netzwerk. Siehe: *Ein Interface für OLSR konfigurieren*.

Einrichtung eine Alias-Schnittstelle für OLSR auf der Shell

1. Alias Interface anlegen

Zunächst muss ein *Alias-Interface* angelegt werden. Dazu wird am Ende von `/etc/config/network` folgendes eingefügt:

```
config interface 'lanolsr'
option proto 'static'
option ifname '@lan'
option ipaddr '10.11.9.100'
option netmask '255.255.192.0'
```

Alternativ kann das auch mit **uci** erledigt werden:

```
uci set network.lanolsr='interface'
uci set network.lanolsr.proto='static'
uci set network.lanolsr.ifname='@lan'
uci set network.lanolsr.ipaddr='10.11.9.100'
uci set network.lanolsr.netmask='255.255.192.0'
```

2. Alias Schnittstelle zur Firewallzone ‘‘freifunk‘ hinzufügen

In `/etc/config/firewall` muss die `freifunk`-Zone bearbeitet und dort das eben angelegte Alias-Interface `lanolsr` für `option network` hinzugefügt werden.

```
config zone 'zone_freifunk'
     option name 'freifunk'
     option input 'REJECT'
     option forward 'REJECT'
     option output 'ACCEPT'
     option masq '1'
     list masq_src 'lan'
     list masq_src 'wireless0dhcp'
     list masq_src 'wireless0ahdhcp'
     option network 'wireless0 wireless0dhcp lanolsr'
```

Auch das ist wiederum mit **uci** direkt möglich:

```
net="$(uci get firewall.zone_freifunk.network)"
uci set firewall.zone_freifunk.network="$network lanolsr"
uci commit firewall
```

> **Warnung:** Bei uci müssen wir hier den Weg gehen, zunächst herauszufinden, welche Netze bereits zur Zone gehören (Die Variable `network` aus der ersten Zeile).

3. OLSR für das Interface aktivieren

Jetzt muss noch *OLSR* für diese neu angelegte Schnittstelle aktiviert werden. Verwende dabei `lanolsr` als Netzwerk. Siehe: *Ein Interface für OLSR konfigurieren*.

4. Dienste neu starten

```
/etc/init.d/network restart
/etc/init.d/olsrd restart
```

Konfiguration des angeschlossenen Rechners

Der angeschlossene Rechner muss eine IP aus dem Mesh-Netzwerk der eigenen Community verwenden. Siehe *IP-Adresse(n) registrieren*.

Zudem muss *OLSR* auf dem Rechner konfiguriert und gestartet werden.

Hinweis: Auf dem Freifunkrouter findet man auf der Seite *Freifunk* → *OLSR* unter *Konfiguration herunterladen* die auf dem Router aktive OLSR Konfiguration. Diese kann sehr gut als Ausgangsbasis für die OLSR-Konfiguration auf dem Rechner verwendet werden.

6.2 Privates WLAN-Netzwerk einrichten

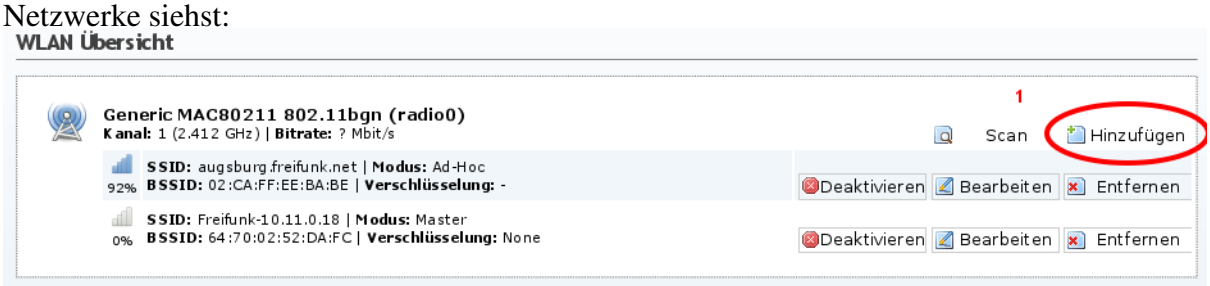
Wenn die WLAN-Hardware des Routers die Einrichtung von *VAP* (Virtuellen Access Points) unterstützt, dann kann zusätzlich zu dem/den Freifunknetz(en) ein **privates WLAN** eingerichtet werden, das Verschlüsselung benutzt und Teil des `LAN`-Netzwerks (siehe *LAN, WAN und Freifunk - Die einzelnen Zonen im Router*) ist.

Hinweis: Soll Verschlüsselung benutzt werden dann muss das Paket `hostapd-mini` installiert werden. Zum Installieren von Paketen siehe *Pakete installieren*.

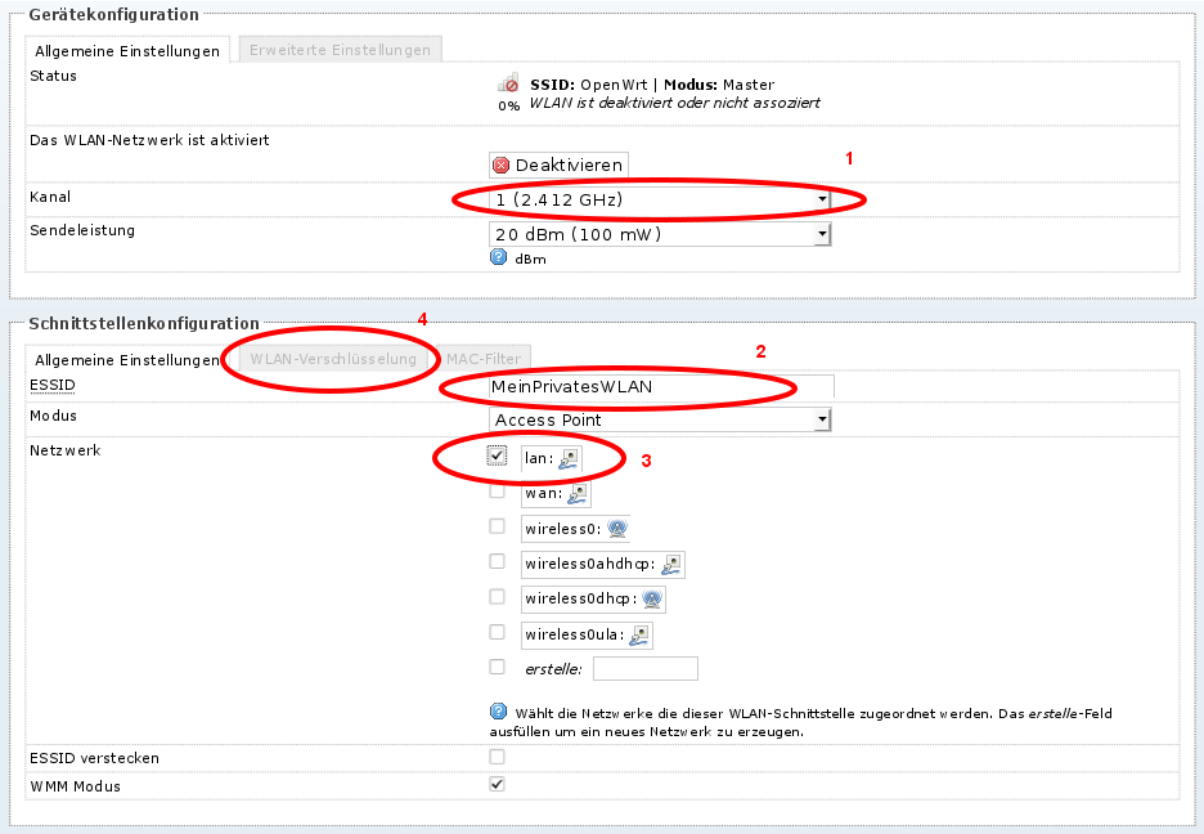
6.2.1 Privates Netzwerk mit LuCI einrichten

Es muss ein weiteres WLAN-Gerät erstellt und dem Netzwerk `LAN` zugeordnet werden.

Gehe dafür zu *Administration* → *Netzwerk* → *Drahtlos* wo du eine Übersicht über die WLAN-Netzwerke siehst:

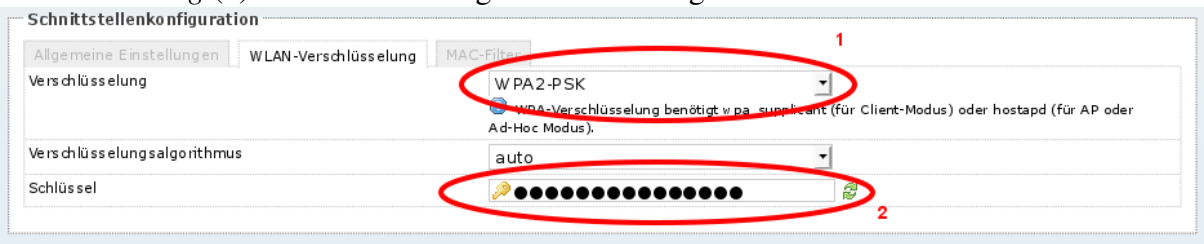


Klicke dort auf *Hinzufügen* (1). Es öffnet sich ein weiteres Fenster mit den Einstellungen für das neue Interface:



Stelle bei *Kanal* (1) zunächst sicher, dass der selbe Kanal wie für die andere(n) SChnittstelle(n) verwendet wird. **Mehrere VAPs können nur auf einem gemeinsamen Kanal betrieben werden.** Ändere anschliessend die *ESSID* (2) des Netzwerks. Wähle aus, dass das Wifi-Interface zum Netzwerk `LAN` (3) gehören soll.

Soll das Netzwerk verschlüsselt werden (empfehlenswert!), dann anschliessend noch zum Tab *Verschlüsselung* (4) wechseln und folgende Einstellungen vornehmen:



Wähle als Verschlüsselungsmethode *WPA-PSK*, *WPA2-PSK* oder *WPA-PSK/WPA2-PSK Mixed Mode* (1). Gib anschliessend bei *Schlüssel* (2) ein ausreichend sicheres Passwort ein.

Nachdem die Einstellungen gemacht wurden, klicke auf *Speichern & Anwenden* um die Einstellungen zu übernehmen.

Du kannst dich nun mit deinem Computer oder Smartphone mit dem verschlüsselten WLAN verbinden.

6.2.2 Privates WLAN auf der Shell einrichten

Es muss die Konfigurationsdatei `/etc/config/wireless` bearbeitet werden. Hier wird amd Ende folgendes eingefügt:

```
config wifi-iface 'radio0_iface_private'
      option device 'radio0'
      option mode 'ap'
      option ssid 'MeinPrivatesWLAN'
      option network 'lan'
      option encryption 'psk2'
      option key 'xxx'
```

Ersetze hierbei wenn notwendig `radio0` mit dem Namen des wifi-device (siehe in der Konfiguration weiter oben im selben File).

Die selben Einstellungen können auch mit **uci** gemacht werden:

```
uci set wireless.radio0_iface_private=wifi-iface
uci set wireless.radio0_iface_private.device=radio0
uci set wireless.radio0_iface_private.mode=ap
uci set wireless.radio0_iface_private.ssid='MeinPrivatesWLAN'
uci set wireless.radio0_iface_private.network=lan
uci set wireless.radio0_iface_private.encryption=psk2
uci set wireless.radio0_iface_private.key='geheimgeheim'
uci commit wireless
```

Anschliessend müssen die WLAN-Interfaces neu gestartet werden, dies kann durch Eingabe von **wifi** erledigt werden.

6.3 USB Speichermedien einbinden

USB-Speichermedien wie externe Festplatten oder USB-Sticks lassen sich an Routern die einen USB-Port verfügen einfach einbinden.

6.3.1 Notwendige Pakete installieren

Grundlegender USB-Suport muss bereits verfügbar sein. Dies sollte jedoch auf den meisten geräten mit USB-Bereits der Fall sein. Wenn nicht dann siehe Basic USB Support im OpenWrt Wiki.

Folgende Pakete müssen für USB-Speichermedien installiert werden, siehe *Pakete installieren*.

- kmod-usb-storage
- block-mount

Warnung: `block-mount` kollidiert zur Zeit mit Datenien, die bereits von `busybox` bereitgestellt werden. Es ist daher notwendig die beiden oben gennanten Pakete mit **opkg install kmod-usb-storage block-mount –force_overwrite** zu installieren. Eine Installation über LuCI ist daher derzeit nicht möglich.

Je nach auf dem externen Medium verwendeten Dateisystem werden noch `kmod-fs-*` Pakete benötigt, die geläufigsten sind:

- kmod-fs-ext4 - für EXT2, EXT3 und EXT4
- kmod-fs-ntfs - für NTFS
- kmod-fs-reiserfs - Für ReiserFS
- kmod-fs-vfat - Für VFAT/FAT32

Wird kmod-fs-vfat verwendet dann muss zusätzlich auch noch eine `Codepage` installiert werden, diese befinden sich in den Paketen `kmod-nls-cp*`. Es ist empfehlenswert, diese Codepages zu installieren:

- kmod-nls-cp437
- kmod-nls-cp850
- kmod-nls-cp852

Ferner wird Support für Charsets benötigt:

- kmod-nls-iso8859-1
- kmod-nls-iso8859-15

6.3.2 Automatisches mounten aktivieren

Nach Installation der benötigten Pakete sind zwei Laufwerke vorkonfiguriert aber nicht aktiviert. Sie verhindern damit ein automatisches Einbinden von externen Laufwerken ins Filesystem. Es ist empfehlenswert diese zunächst zu löschen:

```
uci del fstab.@mount[0]
uci del fstab.@swap[0]
uci commit fstab
```

Per Default werden nun externe Laufwerke sobald sie angeschlossen wurden automatisch an einen Mountpoint in `/mnt/` gemounted, z.B. nach `/mnt/sda1`.

Wenn das nicht funktioniert dann schaue im *System Log* nach ob dort Fehlermeldungen sichtbar sind.

6.3.3 USB-Storage-Geräte konfigurieren und feste Mountpoints zuweisen

Für bessere Kontrolle darüber, wohin Geräte gemounted werden und die Möglichkeit Mountoptionen anzugeben müssen USB-Storage-Geräte manuell konfiguriert werden. Dies kann via LuCI oder auf der Shell des Routers geschehen.

Optionen für das Mounten externer Speichermedien

Folgende Optionen können sowohl mit *LuCI* als auch auf der *Shell* gesetzt werden, um das Einbinden externer Medien zu kontrollieren:

Option LuCI	Option Shell	Beschreibung
Aktiviert	enabled	Bestimmt ob dieses Gerät eingebunden wird
Schnittstelle	device	Gerät/Partition das gemounted werden soll. Wenn *UUID* oder *Label* ebenfalls gesetzt sind, dann werden diese bevorzugt.
Mountpunkt	target	Der Mountpunkt, an den das Gerät im dateisystem des Routers eingebunden werden soll. Übliche Praxis ist, Geräte unterhalb von `/mnt/` einzuhängen.
Dateisystem	fstype	Dateisystem auf dem Datenträger, der eingebunden werden soll, z.B. vfat, ext2, ext3, ext4 oder ntfs. Wird diese option weg bzw. leer gelassen, dann wird versucht den Typ des Dateisystems automatisch zu erkennen.
UUID	uuid	*UUID* des Gerätes oder der Partition, die eingehängt werden soll. Siehe *UUID und Label herausfinden*.
Label	label	Label des Dateisystems, das eingehängt werden soll. Wird gleichzeitig eine *UUID* konfiguriert dann hat die UUID Vorrang. Siehe *UUID und Label herausfinden*.
Mount-Optionen	options	Spezielle Mount-Optionen, die dem **mount**-Kommando mitgegeben werden sollen. Für mögliche Optionen siehe: mount Manpage.

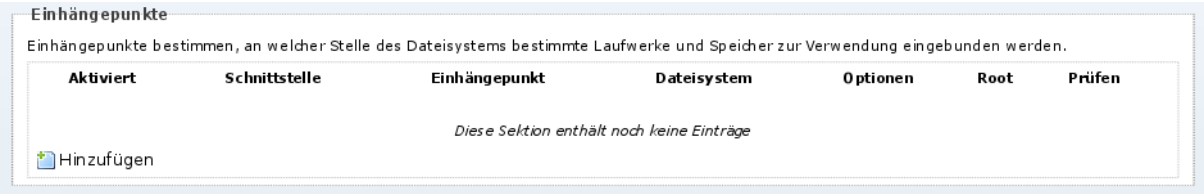
UUID und Label herausfinden

Wenn für einzelne Medien sichergestellt werden soll, dass sie immer an den selben Mountpoint gemountet werden, dann sollte das Medium vorzugsweise anhand seiner *UUID* oder des `Labels` anstelle anhand des `Device`, das sich verändern kann gemounted werden. Um UUID oder Label auf der Shell herauszufinden führt man, während das Speichermedium eingesteckt ist folgendes aus:

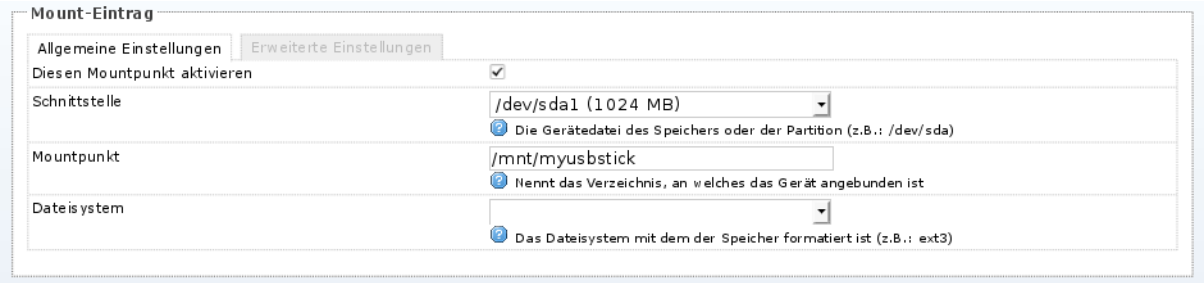
```
root@freifunk:~# blkid
/dev/mtdblock2: TYPE="squashfs"
/dev/sda1: SEC_TYPE="msdos" LABEL="liveusb" UUID="4DE7-F873" TYPE="vfat"
```

Externe Medien in LuCI konfigurieren

Externe Speichermedien und deren Mountpoints können in LuCI unter *Administration* → *System* → *Einhängepunkte* verwaltet werden.



Klicke direkt darunter auf *Hinzufügen*, um ein externes Speichermedium manuell zu mounten. Es öffnet sich eine Seite, auf der das Medium konfiguriert werden muss.



Wähle unter *Schnittstelle* ein Gerät aus. Alternativ kann das Gerät auch im Tab *Erweiterte Einstellungen* anhand seiner *UUID* oder seinem *Label* identifiziert werden. Wähle anschliessend unter *Mountpunkt* den Mountpoint im Dateisystem des Routers aus. Unter *Dateisystem* kann das Dateisystem das sich auf dem externen Medium befindet ausgewählt werden. Wird dies leer gelassen dann wird versucht das Dateisystem automatisch zu erkennen. Unter *Mount-Optionen* im Tab *Erweiterte Einstellungen* können weitere Optionen für den **mount**-Befehl angegeben werden, der dafür zuständig ist, die externen Medien ins Dateisystem einzubinden.

Für eine Erklärung der einzelnen Optionen siehe *Optionen für das Mounten externer Speichermedien*.

Externe Medien auf der Shell konfigurieren

Die Einstellungen für externe Datenträger befinden sich in der Datei `/etc/config/fstab`. Für jeden Mountpunkt wird eine Sektion vom Typ `mount`:

```
config mount 'myusbstick'
      option enabled '1'
      option device '/dev/sda1'
      option target '/mnt/myusbstick'
```

ALternativ können diese Einstellungen auch mit **uci** gemacht werden:

```
uci set fstab.myusbstick=mount
uci set fstab.myusbstick.enabled=1
uci set fstab.myusbstick.device="/dev/sda1"
uci set fstab.myusbstick.target="/mnt/myusbstick"
```

Für eine Erklärung der einzelnen Optionen siehe *Optionen für das Mounten externer Speichermedien*.

6.3.4 Gemountete Geräte anzeigen

In LuCI werden im oberen Bereich unter *Administration* → *System* → *Einhängepunkte* alle gemounteten Geräte angezeigt.

Eingehängte Dateisysteme

Dateisystem	Einhängepunkt	Verfügbar	Belegt
rootfs	/	3.55 MB / 4.13 MB	14% (588.00 KB)
/dev/root	/rom	0.00 B / 2.75 MB	100% (2.75 MB)
tmpfs	/tmp	13.32 MB / 14.25 MB	6% (944.00 KB)
tmpfs	/dev	512.00 KB / 512.00 KB	0% (0.00 B)
/dev/mtdblock3	/overlay	3.55 MB / 4.13 MB	14% (588.00 KB)
overlayfs:/overlay	/	3.55 MB / 4.13 MB	14% (588.00 KB)
/dev/sda1	/mnt/liveusb	943.33 MB / 1023.72 MB	8% (80.39 MB)

Auf der Shell kann das **mount**-Kommando verwendet werden, um diese Informationen anzuzeigen:

```
root@freifunk:~# mount
rootfs on / type rootfs (rw)
/dev/root on /rom type squashfs (ro,relatime)
proc on /proc type proc (rw,noatime)
sysfs on /sys type sysfs (rw,noatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,noatime)
tmpfs on /dev type tmpfs (rw,noatime,size=512k,mode=755)
devpts on /dev/pts type devpts (rw,noatime,mode=600)
/dev/mtdblock3 on /overlay type jffs2 (rw,noatime)
overlayfs:/overlay on / type overlayfs (rw,noatime,lowerdir=/,upperdir=/overlay)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
none on /proc/bus/usb type usbfs (rw,relatime)
/dev/sda1 on /mnt/liveusb type vfat (rw,relatime,[...])
```

In beiden Fällen wird sichtbar, dass das externe Medium `/dev/sda1` auf den Mountpoint `/mnt/liveusb` gemounted ist.

6.3.5 Weiterführende Links

- USB Storage im OpenWrt Wiki

6.4 Eigene Dienste anbieten

Es ist möglich, im Freifunknetz eigene Dienste zur Verfügung zu stellen.

6.4.1 Eigene Webseiten

Lokal auf dem Node

Auf dem Freifunkrouter läuft ohnehin schon ein kleiner Webserver (`uhttpd`), der verwendet werden kann um neben dem LuCI Webinterface auch eigene Inhalte per HTTP zugänglich zu machen.

Seiten direkt über uhttpd ausliefern

Es gibt die Möglichkeit direkt auf dem Node kleinere Seiten abzulegen, indem man sie im Ordner /www unterbringt. Dies erfordert keine Änderung an der Konfiguration des Nodes.

Einfache statische Seite

Beispiel:

```
<html>
  <body>
    Hallo Welt!
  </body>
</html>
```

Den obigen Code als `/www/hallowelt.html` abspeichern und danach im Browser unter der Adresse `http://<nodeip>/hallowelt.html` aufrufen.

CGI-Scripte

Möchte man CGI-Scripte ausführen, dann muss man diese in `/www/cgi-bin` ablegen und ausführbar machen.

Beispiel:

```
#!/bin/sh

echo "Content-type: text/html"
echo ""

echo '
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hallo Welt</title>
  </head>
  <body>
    Hallo Welt!
  </body>
```

```
</html>
'
```

Den obigen Code als `/www/cgi-bin/helloworld.cgi` speichern und ausführbar machen mit:

```
chmod +x /www/cgi-bin/helloworld.cgi
```

Dieses Testscript kann nun im Browser unter

```
http://<nodeip>/cgi-bin/helloworld.cgi
```

abgerufen werden.

Das selbe Script in Lua:

```
#!/usr/bin/lua

print("Content-type: text/html")
print("")

print ([[
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hello World</title>
  </head>
  <body>
    Hallo Welt!
  </body>
</html>
]])
```

Verzeichnisinhalte

In der Standardkonfiguration zeigt uhttpd auch Verzeichnisinhalte von Verzeichnissen an, die keine index-Datei enthalten.

Beispiel:

```
mkdir /www/verzeichnis
echo "hallo" > /www/verzeichnis/hallo.txt
```

Öffnet man im Browser jetzt http://<nodeip>/verzeichnis, dann erhält man folgende Ausgabe:

Auf einem seperaten PC/Server

Soll die Webseite von einem anderen PC bzw. Server ausgeliefert werden, der per Netzwerk mit dem Node verbunden ist gibt es mehrere Möglichkeiten, diese Inhalte im Freifunknetz verfügbar zu machen, siehe dazu: *Angeschlossene Computer im Freifunknetz erreichbar machen*

6.4.2 Dienste ankündigen mit dem Nameservice Plugin

Um eigene Dienste im gesamten Mesh bekannt zu machen kann das *OLSR*-Nameservice Plugin verwendet werden. Dieses sendet in regelmässigen Abständen Informationen über die lokalen Dienste an alle anderen Nodes im Mesh.

Es können nur Service-Ankündigungen für IPs/Adressen verschickt werden, die entweder local verwendet oder als HNA angekündigt werden.

Ist der Service unter einer DNS-Adresse bekannt, dann kann statt einer IP auch diese Adresse in der URL des Dienstes verwendet werden.

Einrichtung

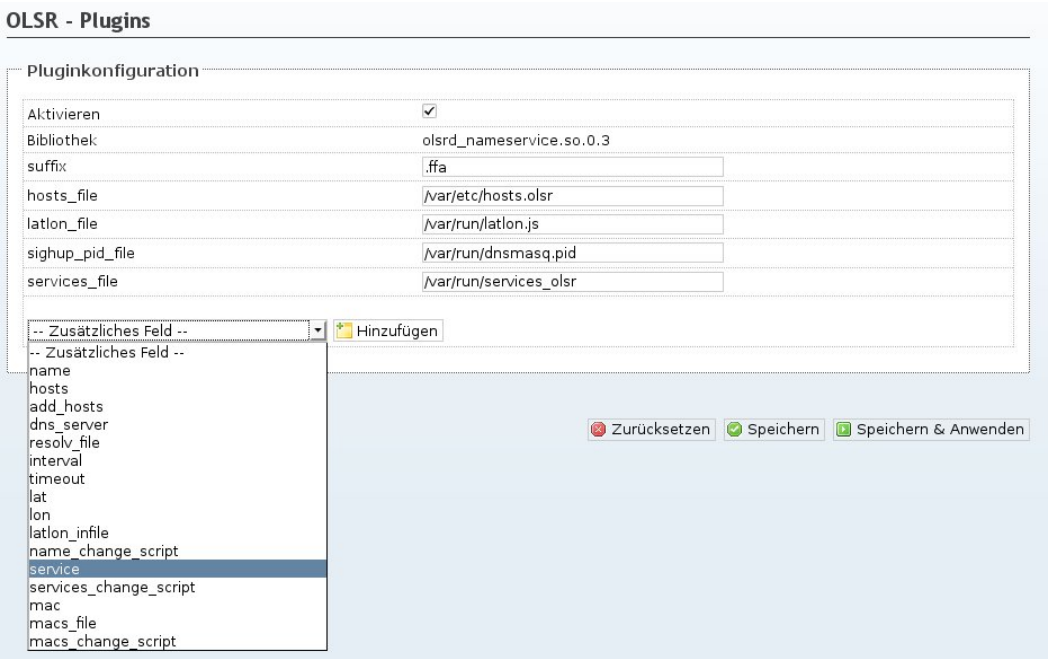
Ziel: Es soll ein Webserver auf dem lokalen Knoten mit der IP 10.11.12.13 angekündigt werden. Der Webserver läuft auf Port 80.

Einrichtung über LuCI

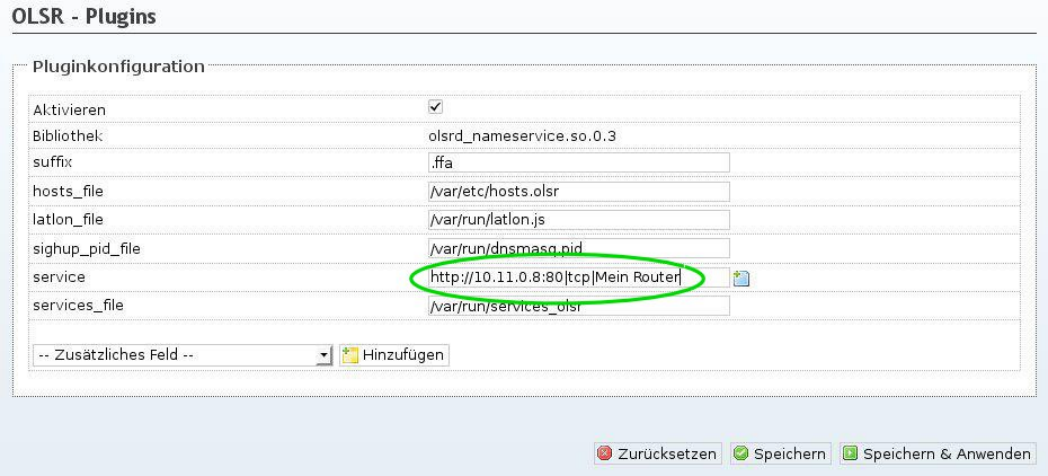
1. Um zu den Einstellungen für das Nameservice Plugin zu kommen: Gehe zu *Dienste -> OLSR -> Plugins*

2. Klicke in der Zeile wo olsrd_nameservice.so.0.3 steht auf “Bearbeiten”

3. Füge unten aus der Auswahlbox eine Option für “Service” hinzu



4. Plugin konfigurieren



Einrichtung mithilfe der Konsole

Um den Webserver auf dem Node `10.11.0.8` Port `80` anzukündigen:

```
uci add_list olsrd.olsrd_nameservice.service=\
  "http://10.11.0.8:80|tcp|Mein Router"
uci commit olsrd
/etc/init.d/olsrd restart
```

Es können auch mehrere Dienste angekündigt werden:

```
uci add_list olsrd.olsrd_nameservice.service=\
  "http://10.11.0.8:80|tcp|Mein Router"
uci add_list olsrd.olsrd_nameservice.service=\
  "ftp://10.11.0.8:21|tcp|Mein FTP Server"
uci commit olsrd
/etc/init.d/olsrd restart
```

Erklärung des Service Strings

Der Aufbau des erwarteten Strings als Option für Service ist recht einfach:

```
<url>:<port>|<Protokoll (tcp oder udp)>|<Beschreibung des Dienstes>
```

Port darf nicht weggeleassen werden!

Ergebnis

Ist alles korrekt eingerichtet dann erscheint im öffentlichen Teil des Webinterfaces auf allen Routern im Mesh unter “Dienste” nach kurzer Zeit die eben eingerichtete Ankündigung für “Mein Router”.

Dienste

Internal services

Url	Protokoll	Quelle
Mein Router	tcp	my own service
Augsburger Blog Aggregator	tcp	10.11.10.29
Bookmarks von soma. Wer will kann da auch nen Account haben.	tcp	10.11.10.10
Internes Wiki	tcp	10.11.10.10

KAPITEL 7

Community-Support

Meshkit erlaubt es einzelnen Communities, eigene Community-Profile zu haben, die bestimmte Defaults vorgeben, welche Meshkit dann benutzt. Ausserdem ist es möglich, pro Community eigene Dateien mit ins Image zu kopieren.

7.1 Community Profile

Jede Community, für die Meshkit Firmwareimages bauen soll, hat ein Community- Profil, in dem einige Standardwerte (wie z.B. Kanal, ESSIDs usw.) stehen.

Diese Vorgaben werden entweder nur im Hintergrund benutzt oder, falls sie durch den Benutzer veränderbar sein sollen, als Standardwerte in Meshkit verwendet.

Für eine Erklärung der einzelnen Optionen siehe: http://wiki.freifunk.net/Kamikaze/Profile

Alle Communityprofile befinden sich im LuCI Repository

7.1.1 Profil für eine neue Community erstellen

Um ein Profil für eine neue Community zu erstellen, schaue dir zunächst die existierenden Communityprofile an. Kopiere dann eins der Profile und passe es für die neue Community an. Danach erstelle ein Ticket oder einen Pull Request auf https://github.com/openwrt/luci/issues, welcher das neue Profil hinzufügt.

7.2 Eigene Dateien pro Community

Eine Community kann eigene Files hinterlegen, die dann genau so mit ins Firmwareimage gebaut werden. Dadurch ist es zum Beispiel möglich, ein eigenes Banner für das Freifunk-Generic-Theme ins Image zu bauen oder Scripte abzulegen (und beim Boot auszuführen), die Funktionen mitbringen, die Meshkit selbst nicht konfigurieren kann.

KAPITEL 8

Dokumentation für Entwickler

8.1 API

Meshkit hat eine API, mit der sich einige Statusinformationen abfragen lassen. Ausserdem können Firmwareimages mit der API erstellt werden.

Alle API-URLs sind erreichbar unter http://<url>/api/json/<api>

8.1.1 status

Gibt den aktuellen Status von Meshkit aus, z.B.

```
{"status": true, "queuedimg": 0, "memfree": "494", "totalimg": 585, "successimg"
```

8.1.2 targets

Liste verfügbarer Targets, z.B.:

```
["ar71xx-generic-attitude_adjustment-35817", "atheros-attitude_adjustment-35864"
```

8.1.3 buildstatus

Informationen zu einem speziellen Build anfragen. Dies benötigt zwingend eine id und passenden Random String (rand).

Beispiel-Aufruf: http://<url>/api/json/buildstatus?id=1234&rand=36a419d3bd7ad460d16b4773cc3b5d5a

Antwort

```
{
  "status": 0,
  "files": [
      "openwrt-ar71xx-generic-tl-wr1043nd-v1-squashfs-sysupgrade.bin",
      "md5sums",
```

```
      "openwrt-ar71xx-generic-vmlinux.elf",
      "openwrt-ar71xx-generic-vmlinux.bin",
      "openwrt-ar71xx-generic-vmlinux.gz",
      "openwrt-ar71xx-generic-root.squashfs-64k",
      "build.log",
      "openwrt-ar71xx-generic-vmlinux-lzma.elf",
      "openwrt-ar71xx-generic-root.squashfs",
      "openwrt-ar71xx-generic-tl-wr1043nd-v1-squashfs-factory.bin",
      "openwrt-ar71xx-generic-uImage-lzma.bin",
      "openwrt-ar71xx-generic-vmlinux.lzma",
      "openwrt-ar71xx-generic-uImage-gzip.bin",
      "openwrt-ar71xx-generic-rootfs.tar.gz"
  ],
  "queued": 0,
  "downloaddir": "http://meshkit.freifunk.net/images//36a419d3bd7ad460d16b4773cc
}
```

Definierte Werte für status: 0: Sucessful 1: Queued, 2: Failed

8.1.4 buildimage

Einen neuen Auftrag zum erstellen eines Firmwareimages erstellen.

Dies benötigt zwingend die Angabe eines targets.

Beispiel-Aufruf: http://<url>/api/json/buildimage?target=ar71xx-generic-attitude_adjustment-35817&profile=TLWR1043&community=augsburg&noconf=1

Antwort:

```
{
  "rand": "af851246c99b82ac8b3cf3fbe30ff100",
  "errors": {},
  "id": 586
}
```

Im Fehlerfall, z.B. bei Angabe ungültiger Variablen, wird ein Dictionary mit Fehlermeldungen zurückgegeben.

KAPITEL 9

Über dieses Handbuch

9.1 Autoren

Folgende Pesonen haben zu diesem Handbuch beigetragen:

- Manuel 'soma' Munz / Freifunk Augsburg

Hinweis: Trage dich hier selbst ein wenn du mitgeholfen hast diese Dokumentation besser zu machen.

9.2 Lizenz

Dieses Handbuch wird unter der Creative Commons by-nc-sa Lizenz veröffentlicht.

9.3 Hilf mit bei der Dokumentation

Eine Dokumentation zu schreiben ist jede Menge Arbeit. Hilfe dabei ist daher gerne gesehen.

9.3.1 Allgemeine Regeln zur Dokumentation

- Dieses Handbuch wird unter der Creative Commons by-nc-sa Lizenz veröffentlicht. Deine Beiträge werden ebenfalls unter dieser Lizenz freigegeben.
- Texte sinnvoll durch Überschriften gliedern
- Bilder: Nur JPG oder PNG verwenden
- Wo möglich verweise auf andere Stellen in der Dokumentation
- Erkläre möglichst gut und dabei so kurz wie möglich
- komplette Screenshots sollten 1024px breit sein

9.3.2 Quellcode dieser Dokumentation

Der Quellcode diese Dokumentation befindet sich auf Github: Meshkit Dokumentation.

Zur Erstellung der Dokumentation wird Sphinx genutzt.

9.4 Installation von Sphinx

Viele Linux Distributionen bieten Sphinx als Paket an, unter Debian kann es installiert werden mit:

```
aptitude install python-sphinx python-pygments
```

Desweiteren benötigt diese Dokumentation ein angepasstes Theme, das mit folgendem Befehl installiert werden kann:

```
easy_install sphinxjp.themes.basicstrap
```

oder alternativ mit

```
pip install sphinxjp.themes.basicstrap
```

9.5 Formatierung mit rst

9.5.1 Überschriften

Überschriften sollen Texte logisch Gliedern. Mehr als 4 Ebenen sind in der Regel nicht sinnvoll.

Level	Unterstreichen mit
1 (Kapitelüberschriften)	=
2	=
3	-
4	^

9.5.2 Codebeispiele

Codeblöcke immer kennzeichnen mit

```
.. code-block:: sh
   while true; do
     ping -n 3 google.de
   done
```

was dann so aussieht:

```
while true; do
  ping -n 3 google.de
done
```

9.5.3 Menüpfade zu Seiten in der GUI

Es gibt speziell für GUI-Pfade ein Label. Dieses sollte für alle Menüpfade verwendet werden, z.B.

```
:menuselection:`Administration --> System --> Administration`
```

Das sieht dann so aus:

Administration → System → Administration

9.5.4 Verweise auf andere Elemente in der GUI

Auch hierfür gibt es ein Label:

```
:guilabel:`Irgendwas in der GUI`
```

was dann so aussieht:

Irgendwas in der GUI

9.5.5 Dateipfade

Pfade zu Dateien und Verzeichnissen werden durch das Label :file: kenntlich gemacht, z.B.

```
:file:`/etc/passwd`
```

was dann so aussieht:

`/etc/passwd`

9.5.6 Kommandos

Kommandos auf der Shell sollten durch das Label :command: kenntlich gemacht werden, z.B.

```
:command:`/etc/init.d/olsrd restart`
```

was dann so aussieht:

/etc/init.d/olsrd restart

9.5.7 Glossar

Abkürzungen wie z.B. DHCP sollten im Glossar (`glossar.rst`) erklärt werden. Um Begriff dann auf die entsprechende Stelle im Glossar zu verlinken wird **:term:** verwendet, z.B.

```
Foo kann mit :term:`DHCP` konfiguriert werden
```

Das sieht im Text dann so aus:

Foo kann mit *DHCP* konfiguriert werden

9.5.8 Tabellen

Um auch im PDF definierte tabellenbreiten zu haben, müssen diese in absoluten Werten angegeben werde. Die Gesamtbreite der Tabelle darf dabei maximal 15cm sein.

Beispiel für eine Tabelle mit 2 Spalten:

```
.. tabularcolumns:: |p{6cm}|p{9cm}|
```

Dies direkt vor der Definition der Tabelle in der .rst Datei einfügen.

9.6 Weiterführende Links zu Sphinx und rst

- Sphinx Dokumentation
- Sphinx Doku bei grund-wissen.de

KAPITEL 10

Glossar

Alias-Interface Schnittstellen können mehrere IP-Adressen haben. Die erste IP der Schittstelle ist die primäre IP. Weitere IP-Adressen auf der Schittstelle sind Alias-IPs. Da bei OpenWrt Aliase als eigene Schnittstellen angelegt werden, sprechen wir hier auch von Alias-Interface.

cron cron ist ein Dienst unter Linux, der periodisch zu definierten Zeiten oder Intervallen Kommandos ausführen kann.

DHCP Dynamic Host Configuration Protokoll. Dies wird benutzt, um Rechnern im Netzwerk automatisch IPv4-Adressen und weitere Einstellungen wie Defaultgateway oder *DNS*-Server zuzuweisen.

DNS Domain Name System

HNA Host/Network Announcement. Damit werden von *OLSR* stellvertretend einzelne Rechner oder Netzwerke im Mesh angekündigt.

JSON JavaScript Object Notation ist ein Datenformat, das für Mensch und Computer einfach lesbar sein soll. Es wird vor allem zum Austausch von Daten zwischen Programmen benutzt. Siehe: http://de.wikipedia.org/wiki/JavaScript_Object_Notation

MAC Weltweit eindeutige Hardwareadresse von Netzwerkschnittstellen. Hat die Form XX:XX:XX:XX:XX:XX.

MID Hat ein Knoten mehrere Interfaces auf denen *OLSR* läuft dann wird eine davon zur primären Schnittstelle erklärt und deren IP als OLSR-Haupt-IP-Adresse verwendet. Die IP-Adressen weiterer Schnittstellen werden als MID bezeichnet.

Netzmaske Die Netzmaske oder auch Subnetzmaske gibt die Grösse von Netzwerken an. Die Netzmaske kann entweder in CIDR-Notation, z.B. /24 oder in der dotted decimal notation, z.B. 255.255.255.0 beschrieben werden. Rechner die in einem von der Netzmaske abgedeckten IP-Bereich liegen werden als direkt erreichbar angesehen. Ein nützliches Tool zum Berechnen von Netzmasken ist **ipcalc**. Mehr zu Netzmasken bei Wikipedia: http://de.wikipedia.org/wiki/Netzmaske

OLSR Optimized Link State Routing. OLSR ist ein Routingprotokoll für Meshnetzwerke, das von der Meshkit Firmware benutzt wird. OLSRd-Homepage.

TFTP Trivial File Transfer Protocol. Dies wird häufig genutzt, um Firmwareimages zum Router zu übertragen.

UCI Unified Configuration Interface. Vereinheitlicht die Konfiguration des OpenWrt Systems. Zur Benutzung von des **uci**-Kommandos auf der Kommandzeile siehe *uci*.

UUID Universally Unique Identifier. Eine eindeutige Kennung für Geräte.

VAP Virtueller Access Point. Sofern die Hardware dies unterstützt, können mehrere voneinander getrennte WLAN-Netzwerke auf einer einzigen WLAN-Netzwerkkarte betrieben werden. Wird unterstützt von Hardware die einen der folgenden Treiber verwendet: `ath5k`, `ath9k`, `madwifi`. Welche Treiber ein Router benutzt findet man in den meisten Fällen über die OpenWrt Table of Hardware heraus.