

**Package OPT\_FREIFUNK für FLI4L**  
**Version 1.1.0**

Author: **Ulrich Wachtel**

Web: **freifunk.wachtelnet.de**

E-Mail: **freifunk@wachtelnet.de**

# Inhaltsverzeichnis

Einleitung.....	4
Features.....	4
Installation.....	5
Hardware- Empfehlung.....	5
Abhängigkeiten:.....	6
Konfiguration.....	7
Schalter zur Paket- Aktivierung.....	7
Schalter zur OLSRD- Aktivierung.....	7
Generelle OLSRD- Einstellungen.....	7
Debug Level:.....	8
Ip Version:.....	8
Start ohne Interface (AllowNoInt): .....	8
Pollrate:.....	8
TC redundancy.....	9
NAT Threshold.....	9
MPR coverage.....	9
LinkQualityFishEye.....	9
Link quality level.....	10
Link quality window size.....	10
LinkQualityDijkstraLimit optimization.....	10
OLSRD Willingness.....	11
Use Hysteresis.....	11
Hysteresis parameter.....	11
IPC Connect Werte.....	12
Max IPC Connections .....	12
IPC Hosts.....	12
Hello interval .....	12
HELLO validity time.....	13
TC interval.....	13
TC validity time.....	13
MID interval.....	13
MID validity time.....	14
HNA interval.....	14
HNA validity time.....	14
LinkQualityMult.....	14
Zusätzliche HNA- Routen.....	15
Interface parameter.....	15
Anzahl der Interfaces.....	15
Name des Interfaces.....	15
BSSID.....	16

RTS.....	16
Fragmentierung.....	16
Sendeleistung.....	17
Weitere Parameter.....	17
Längen- und Breitengrad.....	17
BSSID- Hack.....	18
Freifunk- Graphical User Interface (GUI).....	19
Aktivieren der GUI.....	19
Aktivieren der OLSRD- GUI.....	19
OLSRD- Plugins.....	26
OLSRD dyn_gw_fli4l.....	26
OSLRD nameservice .....	27
OLSRD http-server.....	28
OLSRD dotdraw.....	29
OLSRD pgraph.....	29
Behandlung des Kommandos „iwpriv“.....	29
„HORST“- Tool.....	30
Vorbereitung für den Start.....	30
Start des „HORST“- Tools: .....	30
RRD- Statistik- Erweiterung .....	30
Komfortabler HNA.....	31
ToDo – Was noch fehlt.....	33
Links/ Referenzen.....	34
Danksagung.....	34

## **Einleitung**

Dieses Paket enthält neben dem OLSR- daemon weitere für das Freifunk Projekt wichtige Funktionen. Damit ist ein FLI4L mit diesem Paket nahezu softwarekompatibel mit einem anderen im Freifunk- Netz benutztem Gerät (z.B. WRT54).

So wurden wichtige das Netz stabilisierende Features standardmäßig eingebaut wie z.B. eine feste BSSID oder DijkstraLimit und Fisheye- Optimierung. Ab der Version 1.1.0 gibt es einen Mechanismus, der in einem definierten Zyklus die eingestellte BSSID abfragt und bei Änderung (durch schlechte WLAN- Hardware) die BSSID wieder setzt (s.g. BSSID- Hack).

Weitere Optimierungen wie z.B. die bevorzugte Behandlung der OLSRD- Pakete können relativ einfach mit dem QOS- Paket von FLI4L realisiert werden. Einen kompletten Satz von Beispiel- Konfigurationsdateien für ein FLI4L- Freifunk- Gerät können unter <http://freifunk.wachtelnet.de> bezogen werden. Für HNA aber auch für die anderen Knoten ist mittels „Traffic Shaping“ aus dem OPT\_QOS- Paket auf einfache Art und Weise die vorhandene Bandbreite sinnvoll zu verteilen.

Die Software wurde bisher in einem Netz zusammen mit 1 WRT54GL, 2 Asus WL-HDD2.5, 1 FLI4L getestet. Ein FLI4L fungiert dabei als HNA, ein weiterer als normaler „Nicht- HNA“- Knoten. Besonders positiv machte sich hierbei die festgestellte BSSID auf die Stabilität der Verbindung zum FLI4L- HNA bemerkbar, da die verwendete WLAN- Karte ohne dieses Feature dazu neigte, ab und an ihre eigene BSSID vorzugeben. Im Gegensatz zu anderen Atheros- Karten ist die Langzeitstabilität mit festgesetzter BSSID bei der WNC-0300 hervorragend.

Ab OPT\_FREIFUNK Version 1.1.0 findet die OLSRD- Version 0.5.3 Verwendung.

## **Features**

- OLSRD: DijkstraLimit- Optimierung, Fisheye- Optimierung, LinkQualityMult (Abwertung einer/ mehrerer IP), MAC- Cache (Ab Version 1.1.0)
- OLSRD- dyngwfl4l- Plugin (Ein für FLI4L optimiertes olsrd\_dyngwplain- Plugin)
- OLSRD- Plugin zum Internet- DNS announce und DNS auslesen (Olsrd\_nameservice Plugin)
- OLSRD- Olsrd\_dot\_draw und Olsrd\_pgraph Plugins zur Erstellung von Netzgraphen.
- OLSRD- secure Plugin (nicht benutzt für Freifunk)
- Fest einstellbare BSSID (z.B. „02:CA:FF:EE:BA:BE“)
- BSSID- Hack um die feste BSSID zu erzwingen (Ab Version 1.1.0)

- Einstellbarer Sendepiegel (TXPOWER)
- Einstellbare FRAG und RTS- Parameter
- OLSRD- Defaultwerte (für Freifunk Projekt optimiert)
- Selbst konfigurierbare Freifunk-GUI (Kompatibel mit Freifunk- Firmware)
- Statistik für link quality, neighbour link quality, lost frames (Addon für rrd-tool in der password geschützten minihttp-GUI) (Ab Version 1.1.0)
- HORST- Tool (WLAN- trace tool)
- Offene Freifunk GUI auf Port 80 (Homepage, Statuspage mit Routing, WLAN- Scan und OLSR- Info, Freifunk- Map) (Ab Version 1.1.0)
- Freifunk- GUI ist customisierbar (Sprachversionen, eigene Seiten etc. download aus dem Internet, eigene Links und Bild auf der Homepage, eigene Links auf andere Seiten in der Navigation)
- Optionales „iwpriv“- Kommando für Atheros- Karten
- Paket funktioniert nun mit der stabile 3.0.x FLI4L- Version und der 3.1.x Entwicklungs- Version

## Installation

Das OPT\_FREIFUNK passt auf keine Disketten- Installation.

Daher sollte eine FLI4L- Installation auf HD, CD oder Flashkarte erfolgen.

Es werden folgende Ressourcen benötigt:

Memory: ca. 34 MB

HD/ Flashcard: ca. 4,9 MB

Gerechnet ist hier nicht die Minimalkonfiguration sondern alle unten genannten Pakete für den komfortablen HNA mit diversen Zusatzfunktionen.

## Hardware- Empfehlung

Bisher wurde das Paket mit der folgenden WLAN- Karte getestet:

### **Level one - WNC-0300 (Atheros Chipset)**

Diese Karte erreicht eine Sendefeldstärke von 18 dBm und hat eine gute Empfangsempfindlichkeit. Es werden Modi bis 108 Mbps unterstützt. Mit dem FLI4L- Treiber ist ein sicherer Betrieb im 54 Mbps- Mode möglich. Der Preis der Karte ist OK.

Bisher sehr stabile Zusammenarbeit mit dem alten MADWIFI- Treiber im FLI4L 3.0.x.

## **Abhängigkeiten:**

Für die korrekte Funktion von OPT\_FREIFUNK werden folgende OPT- Pakete benötigt:

<i><b>Packet</b></i>	<i><b>Beschreibung</b></i>
OPT_WLAN	Das Paket enthält auch die „wireless tools“, die von diesem Paket auch benutzt werden. Außerdem bindet es einen WLAN- Treiber Modul ein.
ATH_PCI	Dieser Treiber wird vorzugsweise benutzt, da dieser es in Verbindung mit einer Karte von Atheros ermöglicht, die BSSID fest auf 02:CA:FF:EE:BA:BE zu setzen. Natürlich arbeitet das System auch mit anderen von FLI4L angebotenen WLAN- Treibern, jedoch ist dann mit den Problemen zu rechnen, die bei nicht festgesetzter BSSID auftreten. Siehe BSSID-hack.
OPT_EASYCRON	Das Paket wird benötigt um regelmäßig die Domainnamen im OLSR- Netzwerk in Verbindung mit dem DNS- Plugin aufzufrischen. Ausserdem kann es benutzt werden, um den AP

<i>Packet</i>	<i>Beschreibung</i>
	bei instabilem Verhalten des OLSR- Daemons regelmäßig zu rebooten bzw. den OLSRD neu zu starten.
OPT_HTTPD	Dieses Paket wird ab der Version 1.1.0 benötigt, um die Freifunk- GUI zu realisieren.
OPT_WGET aus dem Tools- Paket	Dieses Paket wird durch die in Version 1.1.0 realisierte Freifunk- GUI benutzt um eigene Websites oder CGI- Skripte zu laden.

## Konfiguration

In diesem Kapitel wird die Konfiguration des OPT\_FREIFUNK erläutert.

### **Schalter zur Paket- Aktivierung**

Aktivieren des gesamten Freifunk- Paketes

Gültige Werte: yes / no

**OPT\_FREIFUNK='yes'**

### **Schalter zur OLSRD- Aktivierung**

Aktivieren des OLSR- daemons. Dieser Schalter wurde eingefügt um in Zukunft OLSRD in Verbindung mit B.A.T.M.A.N. abschalten zu können. Jetzt muss er natürlich erst mal eingeschaltet werden.

Gültige Werte: yes / no

**OPT\_OLSRD='yes'**

### **Generelle OLSRD- Einstellungen**

## **Debug Level:**

Diese Einstellung ist nur für Entwickler geeignet und sollte im Normalfall nicht verändert werden.

Gültige Werte: 0-9.

Standardwert: 0

Mit dem Wert 0 läuft der OLSRD daemon im Hintergrund. Es wird kein Debug- Logfile geschrieben und es erfolgen keine Ausgaben auf die Konsole. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird. Der maximale Output wird im DebugLevel 9 erzielt.

**FREIFUNK\_DEBUGLEVEL=""**

## **Ip Version:**

Hier kann die zu verwendende IP- Version eingestellt werden. Bisher wird jedoch nur IP Version 4 benutzt. Daher den Wert bitte nicht ändern. Das Plugin dyngwfl4l unterstützt bisher auch nur IP V4. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte: 4 oder 6.

Standardwert: 4

**FREIFUNK\_IPVERSION=""**

## **Start ohne Interface (AllowNoInt):**

Diese Variable erlaubt es, den OLSRD auch dann zu starten, wenn noch kein Interface aktiv ist. Dieses Feature ist insbesondere bei Verwendung von „hotwap“ Geräten wie z.B. PCMCIA/ USB WLAN- Module. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte: yes/no.

Standardwert: yes

**FREIFUNK\_ALLOWNOINT=""**

## **Pollrate:**

Die Pollrate wird vom s.g. „event scheduler“ für die Ereignis- Abfrage benutzt. 0.2 bedeutet, dass alle 0,2 Sekunden nach neuen Ereignissen geprüft wird. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte: Sekunde (Positive Gleitpunktzahl)

Standardwert: 0.05

**FREIFUNK\_POLLRATE=""**

## **TC redundancy**

Dieser Parameter beschreibt wie viele Informationen über Nachbarn in der TC Meldung gesendet werden sollen. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte:

0: Sendet nur MPR selectors

1: Sendet MPR selectors und MPR's

2: Sendet alle Nachbarn

Standardwert: 2

**FREIFUNK\_TCREDUNDACY=""**

## **NAT Threshold**

Dieser Parameter ist neu ab OPT\_FREIFUNK Version 1.1.0.

Gültige Werte: Sekunde (Positive Gleitpunktzahl)

Standardwert: 0.9

**FREIFUNK\_NATTHRESHOLD=""**

## **MPR coverage**

Dieser Parameter gibt an wie viele MPR's ein Knoten auswählen soll um einen 2 Hop Nachbarn zu erreichen. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 7

**FREIFUNK\_MPRCOVERAGE=""**

## **LinkQualityFishEye**

Dieser Parameter gibt an, ob der sogenannte „Fisheye- Mechanismus“ für TC- Meldungen benutzt werden soll. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte:

1: bedeutet: benutzen

2: bedeutet: nicht benutzen

Standardwert: 1

**FREIFUNK\_LINKQUALITYFISHEYE=""**

## **Link quality level**

Der Parameter entscheidet ob und wofür die Linkqualität zu verwenden ist. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte:

0: Verwende nicht link quality. OLSRD läuft im "RFC3626 mode"

1: Verwende link quality für die MPR- Selektion

2: Verwende link quality für die MPR- Selektion und Routing

Standardwert: 2

**FREIFUNK\_LINK\_QUALITY\_LEVEL=""**

## **Link quality window size**

Der Wert gibt die Fenstergröße für die Ermittlung der Linkqualität an. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 100

**FREIFUNK\_LINKQUALITYWINSIZE=""**

## **LinkQualityDijkstraLimit optimization**

Im Freifunk- Projekt wird die s.g . „LinkQualityDijkstraLimit“ Optimierung benutzt.

Gültige Werte:

Leer: Bei leerem Wert wird diese Optimierung nicht verwendet.

0 5.0: Optimierung wird verwendet.

Es ist kein Standardwert definiert. Im Freifunk- Netz sollte dieser Wert verwendet werden.

**FREIFUNK\_DIJKSTRALIMIT='0 9.0'**

## **OLSRD Willingness**

Dieser Parameter gibt an, ob der betreffende Knoten bereit ist, als Relay für seine Nachbarn zu arbeiten. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Gültige Werte: 0-7

0: Der Knoten gestattet überhaupt keine Relay- Funktion.

4: Neutrale Option

7: Der Knoten gestattet immer die Relay- Funktion.

Standardwert: 3

**FREIFUNK\_WILLINGNESS=""**

## **Use Hysteresis**

Der Parameter gibt an, ob Hysterese verwendet werden soll oder nicht. Im Freifunk Netz wird die Hysterese nicht verwendet.

Es ist kein Standardwert definiert.

**FREIFUNK\_USE\_HYSTERESIS='no'**

## **Hysteresis parameter**

Diese Parameter haben nur eine Bedeutung, falls Hysterese benutzt wird. Bitte verändere die Werte nur, wenn Du weißt, was Du tust. Die Werte sind positive Gleitpunktzahlen im Intervall von 0.1.

„HystThrLow“ muss kleiner als der Wert „HystThrHigh“ sein. Es existieren keine Standardwerte.

Empfohlene Werte sind:

HystScaling: 0.50

HystThrHigh: 0.80

HystThrLow: 0.30

**FREIFUNK\_HYSTERESIS\_SCALING='0.50'**

**FREIFUNK\_HYSTERESIS\_THRHIGH='0.80'**

**FREIFUNK\_HYSTERESIS\_THRLOW='0.30'**

## **IPC Connect Werte**

### **Max IPC Connections**

Der Parameter definiert, wie viele simultane IPC Verbindungen aufgebaut werden können. Der Wert 0 schaltet den IPC- Mechanismus ab. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 0

**FREIFUNK\_IPC\_MAX\_CONNECTIONS=""**

### **IPC Hosts**

Der Parameter gibt an welche Hosts sich mit dem OLSRD per IPC verbinden dürfen. Standardmäßig darf sich nur der lokale Host (127.0.0.1) verbinden.

Es können hier weitere Rechner hinzugefügt werden. Mehrere Einträge sind erlaubt. Der lokale Rechner 127.0.0.1 muss nicht explizit definiert werden, da dieser bereits intern definiert wurde.

**FREIFUNK\_IPC\_HOST\_N='2'**

**FREIFUNK\_IPC\_HOST\_1='192.168.250.4'**

**FREIFUNK\_IPC\_HOST\_2='192.168.0.4'**

Es können auch ganze Netze freigeschaltet werden. Mehrere Einträge sind erlaubt.

**FREIFUNK\_IPC\_NET\_N='2'**

**FREIFUNK\_IPC\_NET\_1='192.168.250.0:255.255.255.0'**

**FREIFUNK\_IPC\_NET\_2='192.168.1.0:255.255.255.0'**

### **Hello interval**

Über das OLSRD Interface werden „Hello“ Meldungen gesendet. Dieser Parameter bestimmt das Sende Intervall dieser Pakete (in Sekunden/ Fliesskomma- Wert). Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 6.0 (= 6 Sekunden)

**FREIFUNK\_HELLO\_INTERVAL=""**

## **HELLO validity time**

Der OLSRD gibt in der „Hello“ Meldung eine Gültigkeitsdauer (Hello validity time) an (in Sekunden/ Fliesskomma- Wert). Diese muss größer sein als das „Hello“ Intervall. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 108.0 (= 108 Sekunden)

**FREIFUNK\_HELLO\_VALIDITY\_TIME=""**

## **TC interval**

Über das OLSRD Interface werden „TC“ Meldungen gesendet. Dieser Parameter bestimmt das Sende Intervall dieser Pakete (in Sekunden/ Fliesskomma- Wert). Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 4.0 (= 4 Sekunden)

**FREIFUNK\_TC\_INTERVAL=""**

## **TC validity time**

Der OLSRD gibt in der „TC“ Meldung eine Gültigkeitsdauer (TC validity time) an (in Sekunden/ Fliesskomma- Wert). Diese muss größer sein als das „TC“ Intervall. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 324.0 (= 324 Sekunden)

**FREIFUNK\_TC\_VALIDITY\_TIME=""**

## **MID interval**

Über das OLSRD Interface werden „MID“ Meldungen gesendet. Dieser Parameter bestimmt das Sende Intervall dieser Pakete (in Sekunden/ Fliesskomma- Wert). Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 18.0 (= 18 Sekunden)

**FREIFUNK\_MID\_INTERVAL=""**

## **MID validity time**

Der OLSRD gibt in der „MID“ Meldung eine Gültigkeitsdauer (MID validity time) an (in Sekunden/ Fliesskomma- Wert). Diese muss größer sein als das „MID“ Intervall. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 324.0 (= 324 Sekunden)

**FREIFUNK\_MID\_VALIDITY\_TIME=""**

## **HNA interval**

Über das OLSRD Interface werden „HNA“ Meldungen gesendet. Dieser Parameter bestimmt das Sende Intervall dieser Pakete (in Sekunden/ Fliesskomma- Wert). Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 18.0 (= 18 Sekunden)

**FREIFUNK\_HNA\_INTERVAL=""**

## **HNA validity time**

Der OLSRD gibt in der „HNA“ Meldung eine Gültigkeitsdauer (HNA validity time) an (in Sekunden/ Fliesskomma- Wert). Diese muss größer sein als das „HNA“ Intervall. Eine leere Variable bedeutet hier, dass der Standardwert benutzt wird.

Standardwert: 108.0 (= 108 Sekunden)

**FREIFUNK\_HNA\_VALIDITY\_TIME=""**

## **LinkQualityMult**

Dieses Flag dient dazu, einzelne Links über die IP abzuwerten. Dies geschieht mit einem Faktor, nachdem das Feature aktiviert worden ist. Ab Version 1.1.0 sind nun mehrere IP- Adressen möglich.

**FREIFUNK\_LINK\_QUALITY\_MULT\_N='1'**

Hier wird die betroffene IP- Adresse angegeben:

**FREIFUNK\_LINK\_QUALITY\_MULT\_1\_IPADDR='104.15.15.99'**

Hier wird der Faktor für die betreffende IP- Adresse angegeben:

Gültige Werte: Fließkomma- Zahl.zwischen 0.1 und 0.9

**FREIFUNK\_LINK\_QUALITY\_MULT\_1\_QUALITY='0.5'**

## **Zusätzliche HNA- Routen**

Im Normalfall gibt FLI4L als HNA schon eine Default- Route vor, wenn FLI4L als Router arbeitet und mit dem Internet verbunden ist. Dann muss man hier keinen Eintrag angeben. Außerdem werden Default- Routen auch viel besser dynamisch durch das Plugin DYNGWFLI4L ermittelt.

Arbeitet FLI4L nicht als HNA, dann sorgt der OLSR- Deamon dafür, dass die richtigen Routen und auch die Default- Route richtig gesetzt wird.

Diese Option gestattet es darüber hinaus statische Routen anzugeben. Von mir wird dieses Feature derzeit nicht genutzt.

**FREIFUNK\_HNA4\_ROUTE\_N='0'**

**FREIFUNK\_HNA4\_ROUTE\_1='192.168.250.1:255.255.255.0'**

**FREIFUNK\_HNA4\_ROUTE\_2='192.168.1.1:255.255.255.0'**

## ***Interface parameter***

Es können prinzipiell mehrere OLSRD Interfaces definiert werden. In der jetzigen Version werden die auf Schnittstellen bezogenen Interface Parameter für jedes Interface getrennt und die OLSRD- Parameter für alle Interfaces identisch eingerichtet.

## **Anzahl der Interfaces**

Der Parameter gibt an, wieviele Interfaces (Schnittstellenkarten) existieren.

**FREIFUNK\_INTERFACE\_N='2'**

## **Name des Interfaces**

Es sind hier gültige wireless Interfaces des FLI4L zu verwenden.

**FREIFUNK\_INTERFACE\_1='eth1'**

## **BSSID**

Es hat sich im Freifunk- Projekt herausgestellt, dass eine fest definierte BSSID die Stabilität des Netzes deutlich erhöht und ein Netzsplit durch fehlerhafte Nodes ausgeschlossen werden kann. Es können sich nur noch Stationen mit gleicher BSSID verbinden und die festgesetzte BSSID kann von einer anderen Station nicht mehr verändert werden.

Dieser Modus wird vom OPT\_FREIFUNK unterstützt, setzt jedoch zwingend den MADWIFI-Treiber (old) von FLI4L und entsprechende Atheros- kompatible Hardware voraus. Wird ein anderer Treiber verwendet muss ausgetestet werden, ob dieser Parameter damit auch wirkt. Dieser Parameter wird intern umgesetzt auf „iwconfig eth0 ap 02:CA:FF:EE:BA:BE“. Im FLI4L-Development- Paket wird der neue madwifi-ng Treiber verwendet. Mit diesem sollte es auch funktionieren.

Im Berliner Freifunk Netz ist die BSSID „02:CA:FF:EE:BA:BE“ einzustellen. Bitte informiere Dich, welche BSSID in Deinem Netz benutzt wird.

**FREIFUNK\_INTERFACE\_1\_BSSID='02:CA:FF:EE:BA:BE'**

## **RTS**

RTS aktiviert einen s.g. Handshake- Mechanismus, der das Verhalten bei einer großen Anzahl von Nodes verbessert. Im Freifunk Netz wird der Wert 250 verwendet.

**FREIFUNK\_INTERFACE\_1\_RTS='250'**

## **Fragmentierung**

Fragmentierung verringert die Paketgröße und kann das Verhalten bei sehr dichten Netzen mit vielen Kollisionen verbessern. Falls der Wert im Freifunk Netz benutzt werden soll, wird der Wert 512 empfohlen.

Gültige Werte:

leer: keine Fragmentierung auf kleinere Pakete.

512: Dieser Wert wird im Freifunk Netz empfohlen.

**FREIFUNK\_INTERFACE\_1\_FRAG=""**

## **Sendeleistung**

Nicht immer muss der Sender mit der vollen Leistung senden. Z.B. wenn man den Gewinn der Antenne und damit den Gesamtgewinn erhöht kann man die Sendeleistung auf einen vom Gesetzgeber vorgeschriebenen Wert drosseln. Es ist ein numerischer Wert, der die Leistung in mW angibt.

Gültige Werte:

leer: Standardwert der Karte wird nicht verändert.

xxx: Numerischer positiver Wert. Einheit muss nicht angegeben werden. Zahlenwert in mW.

**FREIFUNK\_INTERFACE\_1\_TXPOWER=""**

Achtung: Mitunter sind die einstellbaren Werte der Sendeleistung an den Schnittstellenkarten nach oben hin begrenzt und können nicht beliebig erhöht werden.

Ein zweiter Interface- Eintrag sieht dann so aus:

**FREIFUNK\_INTERFACE\_2='eth2'**

**FREIFUNK\_INTERFACE\_2\_BSSID='02:CA:FF:EE:BA:BE'**

**FREIFUNK\_INTERFACE\_2\_RTS='250'**

**FREIFUNK\_INTERFACE\_2\_FRAG='512'**

**FREIFUNK\_INTERFACE\_2\_TXPOWER=""**

## **Weitere Parameter**

### **Längen- und Breitengrad**

Diese Angaben geben die geografischen Koordinaten des Knotens an. Sie werden von der Freifunk-Map benötigt um den eigenen geografischen Standort dem zentralen Kartenserver mitzuteilen. Ausserdem werden die Koordinaten durch die neue Version des OLSRD- Nameservice Plugins benutzt um den eigenen Standort zu announce.

Die Ermittlung der Koordinaten erfolgt auf der MAP unter <http://map.olsrexperiment.de/> bzw.

<http://map.berlin.freifunk.net/>. In der Karte wird der eigene Standort gesucht. Dann komplett reinzoomen und durch einfachen Klick auf den Standort werden die geografischen Koordinaten angezeigt. Nun können diese mit „Cut& Paste“ in die eigene Konfiguration übernommen werden.

Gültige Werte: Positive Gleitpunktzahl

**FREIFUNK\_LAT='xx.yyyyyyyyyyyyyyy'**

**FREIFUNK\_LON='aa.bbbbbbbbbbbbbbbb'**

## **BSSID- Hack**

Leider hat sich herausgestellt, dass die am Markt verfügbare WLAN- Hardware im ad-hoc Mode nicht immer das macht was sie soll. So sollte der WLAN- Karte der später hinzukommt die BSSID des Nachbarn im gleichen Netz übernehmen. Bei manchen Karten passiert dies, bei manchen nicht. Hier hilft bei manchen Karten, die BSSID definiert zu setzen, was OPT\_FREIFUNK unterstützt (iwconfig eth0 ap 02:CA:FF:EE:BA:BE). Nun gibt es wiederum Hardware, die man zwar explizit setzen kann, die jedoch diese gesetzte BSSID nach ein paar Sekunden wieder vergessen. An dieser Stelle setzt der s.g. BSSID- Hack an. Es wird im Hintergrund im Abstand von 2 Sekunden die BSSID geprüft und falls nötig wieder korrigiert.

Dieser Hack sollte nur aktiviert werden, wenn man exakt dieses Problem hat, da die Hintergrund Routine zusätzliche CPU- Zeit konsumiert.

Gültige Werte: yes / no

**OPT\_FREIFUNK\_BSSIDHACK='no'**

Das Thema der BSSID- Stabilität sowie die Benutzung von Atheros- Karten im Zusammenhang mit den Madwifi- Treibern wurde in der Freifunk- Gemeinde bisher sehr konträr diskutiert. Da ist die Rede von „Einschlafenden“ Karten oder von Karten mit diesem BSSID- Problem, bei dem die BSSID ihren richtigen Wert verliert. Beide Probleme konnte ich mit der Karte WNC-0300 in Verbindung mit dem Madwifi (OLD!) Treiber nicht feststellen. Die Ursache für die Probleme bei anderen Usern kann nur auf die Kombination anderer Karten mit anderen Treibern (z.B. madwifi-ng der Development- Version) erklärt werden. An dieser Stelle besteht letztlich weiterer Testbedarf. Aus diesem Grunde empfehle ich auch nicht die Development- Version von FLI4L einzusetzen, da dort nur der neue madwifi-ng Treiber zur Verfügung steht. Dieser Treiber ist jedoch noch in ständiger Weiterentwicklung. Von anderen FLI4L- Usern, die den alten madwifi- Treiber in der stabilen FLI4L- Version benutzen sind mir bisher auch keine solche Probleme berichtet worden.

## **Freifunk- Graphical User Interface (GUI)**

### **Aktivieren der GUI**

Aktivieren der Freifunk- GUI. Mit diesem Schalter wird die Freifunk- GUI auf Port 80 aktiviert.

Gültige Werte: yes / no

**OPT\_FREIFUNK\_GUI='yes'**

### **Aktivieren der OLSRD- GUI**

Dieser Schalter wurde eingefügt um in Zukunft die Teile der GUI die für OLSRD benutzt werden in Verbindung mit B.A.T.M.A.N. abschalten zu können. Jetzt muss er natürlich erst mal eingeschaltet werden.

OPT\_FREIFUNK\_TXTINFO muss aktiviert werden, wenn die Freifunk- Map oder die OLSR- Status- Anzeige verwendet werden sollen.

Gültige Werte: yes / no

**OPT\_FREIFUNK\_OLRSDGUI='yes'**

### **GUI- Informationen Kontaktseite**

Die Freifunk- GUI enthält eine Kontaktseite. Auf dieser Seite sind die folgenden Informationen: Spitzname, Name, Kontakt- Email, Telefonnummer, Ort des Knotens, Notiz. In der Notiz ist mindestens die IP- Adresse des WLAN- Interfaces einzutragen.

Gültige Werte: Zeichenkette

**FREIFUNK\_GUI\_NICKNAME='Musterli'**

Gültige Werte: Zeichenkette

**FREIFUNK\_GUI\_NAME='Max Mustermann'**

Gültige Werte: Email- Adresse

**FREIFUNK\_GUI\_EMAIL='freifunk@xxx.net'**

Gültige Werte: Zeichenkette

**FREIFUNK\_GUI\_FONE='0123456789'**

Gültige Werte: Zeichenkette

**FREIFUNK\_GUI\_LOCATION='Berlin'**

Gültige Werte: IP- Adresse

**FREIFUNK\_GUI\_NOTICE='104.xx.xx.xx'**

## **GUI- Map- Daten**

Die Freifunk- GUI enthält auch die Freifunk- Map. Für diese Funktion sind einige Daten zu administrieren.

Die Freifunk- Map arbeitet mit einem Map- Server zusammen. Die Adresse des Server- Scripts wird hier definiert:

Gültige Werte: Webadresse

**FREIFUNK\_GUI\_MAPSERVER=<http://www.layereight.de/freifunkmap.php>**

Hier stellt man das Update- Intervall ein:

Gültige Werte: hourly, daily, monthly

**FREIFUNK\_GUI\_MAPINTERVAL='hourly'**

In der Map wird dann folgende IP- Adresse angezeigt:

Gültige Werte: IP- Adresse

**FREIFUNK\_GUI\_MAPIP='104.xx.xx.xx'**

## **GUI- Standardwerte der Homepage**

Die Homepage der Freifunk- GUI enthält Standard- Texte. Diese Texte können hier ausgeblendet und dann ggf. durch eigene Texte ersetzt werden.

Gültige Werte: yes/ no

Anzeige des Textes über dem Bild:

**FREIFUNK\_GUI\_SHOW\_DEFTEXTTOP='yes'**

Anzeige des Textes unter dem Bild:

**FREIFUNK\_GUI\_SHOW\_DEFTEXTDOWN='yes'**

Anzeige des Standardbildes:

**FREIFUNK\_GUI\_SHOW\_DEFPIC='yes'**

Anzeige der Default- Links:

**FREIFUNK\_GUI\_SHOW\_DEFLINKS='yes'**

## **GUI- Seiten- Anzeige**

Die Freifunk- GUI enthält folgende Seiten: Homepage, Statusseite, Kontaktseite, Map- Seite.

Jede dieser Seiten kann einzeln ausgeblendet werden und später ggf. durch eigene Seiten ersetzt werden. Die Homepage kann nicht komplett ausgeblendet werden.

Gültige Werte: yes/ no

Anzeige der Status- Seite:

**FREIFUNK\_GUI\_SHOW\_STATUSPAGE='yes'**

Anzeige der Kontakt- Seite:

**FREIFUNK\_GUI\_SHOW\_CONTACTPAGE='yes'**

Anzeige der Map- Seite (Freifunk- Map):

**FREIFUNK\_GUI\_SHOW\_MAPPAGE='yes'**

## **GUI- Status- Seiten- Bestandteile**

Die Statusseite enthält die Bestandteile: Übersicht, Routen, WLAN Scan, OLSR- Statusseite. Diese können bis auf die Übersicht einzeln ein-/ ausgeblendet werden.

Gültige Werte: yes/ no

Anzeige der Routen:

**FREIFUNK\_GUI\_SHOW\_ROUTING='yes'**

Anzeige des WLAN- Scanners:

**FREIFUNK\_GUI\_SHOW\_WLANSCAN='yes'**

**Achtung:** Der WLAN- Scanner funktioniert nicht mit jeder Hardware/ Treiber. Z.B. mit der WNC-0300 und dem atheros-alt funktioniert der Scanner nicht!

Anzeige der OLSRD- Informationen:

**FREIFUNK\_GUI\_SHOW\_OLSRINFO='yes'**

Die Übersichtsseite enthält Links, die weitere Informationen über den Router offenbaren. Diese können ebenfalls aktiviert/ verborgen werden. Dies sind: der Bootlog, ein IPTABLES- Display, die Interfaces und die gerade aktiven Verbindungen.

Gültige Werte: yes/ no

Anzeige des Bootlog:

**FREIFUNK\_GUI\_SHOW\_BOOTLOG='yes'**

Anzeige der IPTABLES:

**FREIFUNK\_GUI\_SHOW\_IPNAT='yes'**

Anzeige der Interfaces:

**FREIFUNK\_GUI\_SHOW\_INTERFACES='yes'**

Anzeige der gerade aktiven Verbindungen:

**FREIFUNK\_GUI\_SHOW\_CONNECTIONS='yes'**

## **Eigene GUI- Seiten und Seiten- Bestandteile (Customization)**

Im Gegensatz zur GUI eines WRT kann die Freifunk- GUI im FLI4L völlig frei verändert werden. Mit den bereits besprochenen Schaltern ist es möglich, die vordefinierten Seiten bzw. Bestandteile aus dem Display zu unterdrücken. An deren Stelle kann nun eigener Content gesetzt werden. Eigener Content wird immer von einer Default- URL im Internet auf den Router bei jedem Bootvorgang heruntergeladen, und dort abgespeichert. Voraussetzung ist natürlich, dass genug Speicherplatz zur Verfügung steht. Als Beispiel ist z.B. eine englische Sprachversion des Default- Content in einem lokalen Download- Verzeichnis auf dem Router (<http://127.0.0.1/download/>) abgelegt. Diese Basis- Download- Adresse (Download- URL) muss festgelegt werden:

**FREIFUNK\_GUI\_PAGE\_DLURL='http://127.0.0.1/download'**

Beim Zugriff auf einen eigenen Webservice im Internet mit eigenem Content sieht es dann so aus:

**FREIFUNK\_GUI\_PAGE\_DLURL='http://www.eigen-server.de/my-content'**

Diese URL gibt das Verzeichnis im Internet an, in dem sich der eigene Content befinden muss.

Der eigene Content wird in einem Verzeichnis auf dem Router abgelegt. Dies ist in der Regel ein Verzeichnis im RAM (z.B. /tmp):

**FREIFUNK\_GUI\_PAGE\_LOCAL='/tmp'**

Natürlich kann sich das lokale Verzeichnis auch auf einer Festplatte befinden (/data):

**FREIFUNK\_GUI\_PAGE\_LOCAL='/data'**

Die eigenen Seiten können statische html oder dynamische shell- cgi- Skripte. Referenzierte Bilder usw. werden nicht in den Router geladen und sind entsprechend zu verlinken!

HTML- Seiten haben auf dem Router die Endung .html !

CGI- Skripte haben auf dem Router die Endung .cgi !

Damit die Dateien nun wirklich auf den Router geladen werden und nicht nur „gebrowsed“ werden müssen die Dateinamen beim Abspeichern der Dateien auf dem Server definiert geändert werden.

HTML- Seiten haben auf dem Download Server die Endung .html.txt !

CGI- Scripte haben auf dem Download Server die Endung .cgi.txt !

Die .txt Endung wird nach dem Transfer dann automatisch abgeschnitten.

**Beispiel:** Eine eigene Datei auf '<http://www.eigen-server.de/my-content>' heisst „MyScript.cgi.txt“.

Es wird nun festgelegt, dass nur eine eigene Seite hinzugefügt werden soll:

**FREIFUNK\_GUI\_PAGE\_N='1'**

Nun wird der Name des Links festgelegt, unter dem das CGI- Script in Freifunk- GUI- Menü erscheinen soll:

**REIFUNK\_GUI\_PAGE\_1='MyPage'**

Nun wird dem FLI4L der Name der Datei mitgeteilt. **Hierbei wird jedoch die .txt Endung weggelassen! Es zählt der Name nach dem Transfer!**

**FREIFUNK\_GUI\_PAGE\_1\_FILE='MyScript.cgi'**

Nun ist noch die Reihenfolge der Links in der Navigationsleiste festzulegen. Order= 50 heißt dass der Link an erster Stelle zu sehen sein wird, 51 an zweiter und so weiter. Auf diese Art und Weise kann die Reihenfolge der Links beliebig festgelegt werden.

Gültige Werte: 50-99

**FREIFUNK\_GUI\_PAGE\_1\_ORDER='50'**

Zu guter Letzt ist für den Link festzulegen, wie die Seite im Browser geöffnet werden soll. Dies entspricht der HTML- Syntax:

" , um kein Target im Verweis zu benutzen,

**FREIFUNK\_GUI\_PAGE\_1\_TARGET=''**

'\_blank' , um den Verweis in einem neuen Fenster zu öffnen,

'\_self' , um den Verweis im aktuellen Fenster zu öffnen,

'\_parent' , um bei verschachtelten Framesets das aktuelle Frameset zu sprengen,

'\_top' , um bei verschachtelten Framesets alle Framesets zu sprengen.

**FREIFUNK\_GUI\_PAGE\_1\_TARGET='\_blank'**

Zusätzlich zu eigenen Seiten und Scripten, die auf den Router transferiert werden ist es auch möglich, Seiten im Internet in der Navigationsleiste zu verlinken.

Dazu wird die Anzahl der Links festgelegt:

**FREIFUNK\_GUI\_PAGELINK\_N='1'**

Nun wird der Name des Links festgelegt, unter dem die Seite aus dem Internet in Freifunk- GUI-Menü erscheinen soll:

**FREIFUNK\_GUI\_PAGELINK\_1='Web FTP'**

Nun wird die URL für den Link festgelegt:

**FREIFUNK\_GUI\_PAGELINK\_1\_URL='https://ssl.wachtelnet.com/wachtelnet\_de/cgi-bin/ftp/ftp.pl/?bp=init'**

Hier ist wieder die Reihenfolge der Links in der Navigationsleiste festzulegen.

Gültige Werte: 50-99

**FREIFUNK\_GUI\_PAGELINK\_1\_ORDER='54'**

Nun ist noch das Target für den Browser festzulegen. Siehe FREIFUNK\_GUI\_PAGE\_1\_TARGET.

**FREIFUNK\_GUI\_PAGELINK\_1\_TARGET='\_blank'**

Neben komplett eigenen Seiten können auch Teile der Standard- Homepage ausgetauscht oder eigene Fragmente hinzugefügt werden:

Text vor dem Bild, Text nach dem Bild, das Bild selbst.

So bekommt man sein eigenes Bild nebst dem HTML- „alternativen Text“. In diesem Beispiel heißt das Bild „introcustom.jpg“ und liegt ebenfalls auf dem Content Server: '<http://www.eigen-server.de/my-content>'.

**FREIFUNK\_GUI\_PIC\_FILE='introcustom.jpg'**

Der alternative Text:

**FREIFUNK\_GUI\_PIC\_ALT='This is my own picture'**

Eigener Text vor dem Bild wird in einer Textdatei definiert und liegt ebenfalls auf dem Content Server: '<http://www.eigen-server.de/my-content>'. Hierbei findet keine Namensänderung nach dem Transfer statt.

**FREIFUNK\_GUI\_TEXT\_BEFORE\_FILE='index-before-gb.txt'**

Eigener Text nach dem Bild wird in einer Textdatei definiert und liegt ebenfalls auf dem Content Server: '<http://www.eigen-server.de/my-content>'. Hierbei findet keine Namensänderung nach dem Transfer statt.

**FREIFUNK\_GUI\_TEXT\_AFTER\_FILE='index-after-gb.txt'**

Auf der Homepage können neben den Standard- Links (oder statt dessen) eigene Links plaziert

werden:

Dazu wird die Anzahl der Links festgelegt:

**FREIFUNK\_GUI\_LINK\_N='1'**

Nun wird der Name des Links festgelegt, unter dem die Seite aus dem Internet in der Linkliste auf der Homepage erscheinen soll:

**FREIFUNK\_GUI\_LINK\_1='Web FTP'**

Nun wird die URL für den Link festgelegt:

**FREIFUNK\_GUI\_LINK\_1\_URL='https://ssl.wachtelnet.com/wachtelnet\_de/cgi-bin/ftp/ftp.pl/?bp=init'**

Nun ist noch das Target für den Browser festzulegen. Siehe FREIFUNK\_GUI\_PAGE\_1\_TARGET.

**FREIFUNK\_GUI\_LINK\_1\_TARGET='\_blank'**

## ***OLSRD- Plugins***

### **OLSRD dyn\_gw\_fli4l**

Dieses OLSRD- Plugin ist ein speziell für FLI4L angepasst worden. Es ermittelt automatisch den Zustand der pppoe- Verbindung und steuert damit das Senden der HNA- Pakete. Damit erhält das OLSR- Netzwerk automatisch eine Information darüber, ob der HNA noch zur Verfügung steht oder nicht. Es werden alle Dialmodi von FLI4L berücksichtigt.

Das Plugin arbeitet nur mit IP- Version 4.

Neben PPPOE als Internet- Zugangsart wird nun auch ein indirekter Internet- Zugang über die default- Route unterstützt.

Schalte hier auf „yes“, wenn Dein FLI4L als HNA dienen soll. Schalte auf „no“, wenn Dein FLI4L kein HNA ist.

Das Plugin dyn\_gw wird nicht mehr unterstützt.

Gültige Werte: yes/ no

**OPT\_FREIFUNK\_DYNGWFLI4L='yes'**

Im normalen Modus wird vom Plugin der Status der ppp- Verbindung des FLI4L überwacht und in Abhängigkeit davon HNA angekündigt. Falls der Router jedoch nicht über PPP sondern über eine IP- Verbindung mit dem Internet verbunden ist, ist dieser Modus nicht hilfreich. Dafür gibt es jetzt den neuen „Ping- Mode“. Dabei wird regelmässig an die angegebenen URL ein Ping abgesendet. Ist dieser erfolgreich, so wird von einer vorhandenen Internet- Verbindung ausgegangen und dann

HNA angekündigt. Der Mode wird mit folgender Variable ein-/ ausgeschaltet:

Gültige Werte: yes/ no

**FREIFUNK\_DYNGWFLI4L\_PINGMODE='no'**

Die Vereinbarung der anzupingenden Hosts erfolgt so:

**FREIFUNK\_DYNGWFLI4L\_HOST\_N='2'**

**FREIFUNK\_DYNGWFLI4L\_HOST\_1='google.de'**

**FREIFUNK\_DYNGWFLI4L\_HOST\_2='fli4l.de'**

Das Ping- Intervall wird in Minuten angegeben:

**FREIFUNK\_DYNGWFLI4L\_PINGTIME='1'**

## **OSLRD nameservice**

Dieses OLSRD- Plugin dient dazu Hostnamen im OLSR- Netzwerk zu veröffentlichen als auch dem eigenen System verfügbar zu machen. Zusätzlich kann mit dem Plugin auch ein HNA- DNS- Name im OLSR- Netzwerk veröffentlicht werden. Ab Version 1.1.0 werden nun auch die geografischen Koordinaten veröffentlicht.

Hintergrund: in jedem FLI4L werkelt ein „dnsmasq“ als Nameserver, der im lokalen Netzwerk in der Regel als Nameserver Cache dient und von den Clients als Nameserver angegeben wird.

Das Plugin liefert zum einen den eigenen Rechnernamen in das OLSR- Netzwerk aus.

Zum anderen werden die empfangenen Namen in einer Datei (var/run/hosts\_olsr) eingetragen, die der dnsmasq mit Hilfe eines Cron- Jobs regelmäßig alle 55 Minuten ausliest. Damit sind auch im lokalen Netz die im OLSR- Netzwerk veröffentlichten Rechnernamen bekannt und können verwendet werden. Dabei findet der NS\_Suffix Verwendung, um eineindeutige Namen im OLSR- Netzwerk sicherzustellen.

Aktivieren des Plugins:

**OPT\_FREIFUNK\_NAMESERVER='yes'**

Gültige Werte: yes/ no

Suffix, der allen empfangenen Namen angehängt wird:

**FREIFUNK\_NS\_SUFFIX='.olsr'**

Prinzipiell kann man mit diesem Plugin auch DNS- Nameserver Adressen ausliefern, die zum

Auflösen von Internet- Namen dienen können. Zum Beispiel die eigene Adresse, wenn man selbst HNA ist und der eigene dnsmasq Internet- Namen auflösen kann.

Dies kann man mit NS\_ANNOUNCE\_DNS und NS\_DNS bewerkstelligen.

FREIFUNK\_NS\_ANNOUNCE\_DNS ist der Schalter und aktiviert die Auslieferung einer DNS- Nameserver- Adresse und FREIFUNK\_NS\_DNS ist die entsprechende IP- Adresse.

Man sollte FREIFUNK\_NS\_ANNOUNCE\_DNS immer auf „no“ setzen, wenn man kein HNA ist.

**FREIFUNK\_NS\_ANNOUNCE\_DNS='no'**

**FREIFUNK\_NS\_DNS='104.15.15.15'**

Allerdings sollte man sich vergegenwärtigen, dass damit die empfangenden Stationen im OLSR- Netz, die ausschließlich diese dynamische Information benutzen, Probleme bekommen, wenn der entsprechende HNA einmal ausfällt. Dann wären zwar noch genügend andere Routen da, aber der DNS wäre nicht mehr verfügbar. Da das Feature optional ist kann man nicht davon ausgehen, dass alle HNA auch dieses verwenden und damit ist der Ausgang offen. Daher empfehle ich hier statische Einträge von Internet- DNS für die DNS- Server zu verwenden. Diese werden wie bisher im „base“- package eingetragen.

Aus all diesen Gründen habe ich auch in diesem Plugin darauf verzichtet die Option "resolv-file" zu verwenden, die das File /etc/resolv.dnsmasq angibt. Diese Datei enthält normalerweise die DNS- Server- Adressen und würde durch das Plugin überschrieben.

## **OLSRD http-server**

Das Plugin wird ab Version 1.1.0 nicht mehr unterstützt.

## **OLSRD Secure**

Das Feature gestattet es Signatur- Meldungen im OLSR- Netz zu verschicken und somit gesicherte routing domains zu schaffen. Dafür muss ein entsprechendes Keyfile auf der Harddisk gespeichert werden.

Dieses Feature wird im Freifunk- Netz nicht verwendet!

Einschalten des Features:

Gültige Werte: yes/ no

**OPT\_FREIFUNK\_SECURE='no'**

Lage des Keyfiles:

**FREIFUNK\_SECURE\_KEYFILE='/boot/data/keyfile'**

## **OLSRD dotdraw**

Das Plugin kann verwendet werden um Topologie Graphen zu erstellen. Mittels eines „Perl“-Scripts, „graphviz“ und „imagemagick“ können auf einem abgesetzten Rechner die Graphen erzeugt werden.

Das Script ist verfügbar unter:

[WWW] <http://meshcube.org/nylon/utils/olsr-topology-view.pl>

Aktivieren des Features:

Gültige Werte: yes/ no

**OPT\_FREIFUNK\_DOTDRAW='yes'**

Zugelassener Host:

**FREIFUNK\_DOTDRAW\_HOST='192.168.250.4'**

Zugelassener Port:

**FREIFUNK\_DOTDRAW\_PORT='2004'**

## **OLSRD pgraph**

Dies ist ein weiteres Plugin zur Erstellung von Topologie Graphen.

Aktivieren des Features:

Gültige Werte: yes/ no

**OPT\_FREIFUNK\_PGRAPH='yes'**

## ***Behandlung des Kommandos „iwpriv“***

Mitunter ist es erforderlich, spezielle private Kommandos an die wireless Schnittstellen auszugeben (iwpriv). Z.B. ist es bei Atheros- Karten möglich, den Mode auf 11b festzustellen. Dies erfolgt in der Regel mit dem Shell- Kommando: „iwpriv eth1 mode 2“

Dieses Kommando kann man automatisch beim Startup des Routers ausführen lassen, indem man folgendes definiert:

**FREIFUNK\_IWPRIV\_N='1'**

**FREIFUNK\_IWPRIV\_1\_INTERFACE='eth1'**

**FREIFUNK\_IWPRIV\_1\_COMMAND='mode 2'**

Somit können beliebige private Kommandos definiert werden.

## **„HORST“- Tool**

Das so genannte „Horst“- Tool ist ein komfortables WLAN- Scan- Tool und kann z.B. gut verwendet werden, um die Sendeleistungen der Nachbarstationen zu ermitteln.

HORST bedeutet: „Horsts OLSR Radio Scanning Tool“.

### ***Vorbereitung für den Start***

Die Schnittstelle muss in den Monitormode gebracht werden, um das „HORST“- Tool aktivieren zu können.

Achtung: Der Monitormode funktioniert nicht mit jeder WLAN- Karte.

Die Aktivierung des Monitormodes erfolgt mit dem folgenden Befehl:

***„iwconfig eth1 mode monitor “***

***ACHTUNG: Bei Atheros- Karten ist in Verbindung mit dem alten Atheros- Treiber kein normaler Traffic mehr möglich, wenn der Monitormode aktiv ist.***

Daher muss nach Nutzung von „HORST“ der Modus wieder zurück auf „Ad-Hoc“ gestellt werden.

Dies geschieht mit: ***„iwconfig eth1 mode ad-hoc“***

### ***Start des „HORST“- Tools:***

Gib einfach ein: ***„horst“***.

Nun startet das „Horst“- Tool und kann verwendet werden.

## **RRD- Statistik- Erweiterung**

Ab Version 1.1.0 ist es nun auch möglich, den Router grafische Statistiken des WLAN- Interfaces schreiben zu lassen.

Als Vorbedingung ist das RRD- Tool zu installieren (OPT\_RRDTOOL)..

Bisher ist es möglich, Link Qualität (LQ), Nachbar Link Qualität (NLQ) und Lost Packets (Lost) mitzuschreiben. Somit kann die Qualität ausgewählter Links überwacht werden.

Um das Feature zu aktivieren ist in der Datei „**/config/rrdtool.txt**“ folgendes einzutragen.

RRDTOOL\_N muss erhöht werden. Angenommen N ist dann gleich 5 wird folgendes hinzugefügt:

Aktivierung des OLSR- Plugins:

**RRDTOOL\_5\_SOURCE='olsr'**

Kommentar für die Lasche in der GUI:

**RRDTOOL\_5\_COMMENT='olsr'**

Und nun erfolgt die Angabe der Funk- Nachbarn, die zu überwachen sind. Die IP- Adressen sind mit Leerzeichen zu trennen:

**RRDTOOL\_5\_OPTIONS='104.xx.xx.xx 104.xx.xx.yy 104.xx.xx.zz'**

Das Intervall gibt an, in welchem Abstand (in Minuten) die Daten aufgefrischt werden.

**RRDTOOL\_5\_VALUE\_INTERVAL='1'**

Dieses Intervall (in Minuten) gibt an, wann die Graphen geschrieben werden. Beim Wert 0 werden die Graphen nur dann erzeugt, wenn der Browser diese anfordert. Das spart CPU.

**RRDTOOL\_5\_GRAPH\_INTERVAL='0'**

Für die Generierung der Daten ist ein Verzeichnis anzugeben. Wichtig ist natürlich bei RRD, dass immer genug Speicher zur Verfügung steht. Falls eine Festplatte vorhanden ist, kann diese verwendet werden. Eine CF- Card sollte aber nicht benutzt werden, da diese nur eine begrenzte Anzahl von Schreibzyklen hat und dann kaputt geht. Empfohlen wird /var/run/rrdtool/olsr.

**RRDTOOL\_5\_GRAPH\_PATH='/var/run/rrdtool/olsr'**

## **Komfortabler HNA**

Für einen komfortablen FLI4L- HNA kommen bei mir folgende Pakete zum Einsatz:

<i>Paket</i>	<i>Beschreibung</i>
OPT_FREIFUNK	Dieses Paket
OPT_HTTP_LEASES	Anzeige der DHCP- Leases in Mini HTTP

<i>Paket</i>	<i>Beschreibung</i>
OPT_ACCOUNTING	Interface Accounting
OPT_WLAN	WLAN- Treiber und Tools
ATH_PCI	ATHEROS- Treiber
OPT_EASYCRON	CRON- Daemon
OPT_HTTPD	HTTP- GUI
OPT_WGET aus dem Tools- Paket	Zeilenorientierter Browser
OPT_DNS	DNS- Server (DNSMASQ)
OPT_DNSDHCP	DHCP- Server (DNSMASQ)
IMOND	Für PC- Tray- Tool (IMONC)
OPT_CGICONFIG2	Tool zum Blocken / Freischalten von Clients und Ports für GUI
OPT_CRONY	Timeserver
OPT_HD	OPT für eine HD/CD- Installation
OPT_MC_3	Midnight Commander
OPT_OPEN_VPN	Open VPN Paket
OPT_POPTOP	PPTP- VPN
OPT_PPP	PPP für DSL und Bluetooth- Verbindung
OPT_QOS	Quality Of Service
OPT_SSHD	SSH- Daemon
OPT_WOL	Wake On LAN- Paket
OPT_USB	USB- driver für Memorystick/ Bt- stick...
OPT_BLUEZ	Bluetooth – Server für Mobile
OPT_SYMBSPOT	Packet für Symbian Handys
OPT_DSL (Nur HNA)	Anschluss an DSL
OPT_DYNDNS (Nur HNA)	DYNDNS- Paket

## ToDo – Was noch fehlt

Leider konnten in der Version 1.1.0 nicht alle angekündigten Erweiterungen realisiert werden. Dafür wurden jedoch andere interessante Dinge realisiert.

Freifunk- GUI	OK
freifunk-setbssid Tool zum Abfragen/ Verändern der Uptime	Offen, Analyse, ob das noch benötigt wird...
LinkQualityMult für mehrere IP's	OK
Dokumentation aller FLI4L- Konfigurations-Dateien	Hier wird es eine Beschreibung der Firewall-Konfiguration geben.
Tauschbörsen Filter	Geplant für 1.2.0
MADWIFI-NG Treiber	Eventuell muss hier überlegt werden, statt des madwifi-ng- Treibers in der Stabile FLI4L- Version lieber den alten madwifi- Treiber für die Dev- FLI4L zur Verfügung zu stellen (Stabilität)
WAS EUCH NOCH SO EINFÄLLT	Auf Anregung von Euch wurden folgenden Features in die 1.1.0 implementiert: <ul style="list-style-type: none"> <li>● OLSRD Upgrade auf V. 0.5.3</li> <li>● Freifunk- Map</li> <li>● iwpriv</li> <li>● RRD- Tool: LQ, NLQ, Ping</li> <li>● Kompatibilität mit FLI4L 3.0.x und 3.1.x</li> </ul>

Was ist für die Version 1.2.0 geplant:

Blacklist/ Whitelist- Feature	Prio 1
Tauschbörsen Filter IPP2P	Prio 1
Weitere RRD- Tool- Parameter	Prio 1
Madwifi- Treiber	Analyse, ggf. Treiber, Prio 2
WAS EUCH NOCH SO EINFÄLLT	

Was ist für die Version 2.0.0 geplant:

B.A.T.M.A.N.	Prio 1
WAS EUCH NOCH SO EINFÄLLT	

## Links/ Referenzen

<i>Link</i>	<i>Beschreibung</i>
<a href="http://freifunk.net">http://freifunk.net</a>	Die deutsche Freifunk- Seite
<a href="http://olsrexperiment.de/">http://olsrexperiment.de/</a>	Die Freifunkseite der Berliner
<a href="http://madwifi.org/">http://madwifi.org/</a>	Die Seite des freien LINUX- Treibers für Atheros WLAN- Karten
<a href="http://freifunk.wachtelnet.de">http://freifunk.wachtelnet.de</a>	Beispielkonfigurationen, Quellen

## Danksagung

- Als erstes vielen Dank an meine Familie, die immer wieder Verständnis für mein Hobby aufbringt.
- Vielen Dank an Patrick Grimm, der nun sowohl in der Implementierung als auch beim Test der neuen Version großen Anteil hatte. Die RRD- Erweiterung, LinkQualityMult, als auch große Teile der FLI4L 3.1.x- GUI stammen von ihm.
- Recht herzlichen Dank an Sven-Ola Tuecke, dem Entwickler der Freifunk- Firmware für WRT54, der mir auch in dieser Version die Feature des neuen OLSRD erklärt hat.
- Vielen Dank auch an all die anderen, die mir mit Anregungen und Tests von Vorab-Versionen eine unschätzbare Hilfe gegeben haben.