

Mikrotik, Metarouter und eine (einfache) Homepage – OE7BSH

SD-Karten sind nicht für die Ewigkeit gebaut und so passiert es immer wieder, dass in einem Raspberry die SD-Karte „schlapp“ macht und damit auch die dort gespeicherte Homepage. Bei uns immer wieder mal passiert, sodass wir uns nach Alternativen umgeschaut haben.

Nachdem am Standort OE7XXR überwiegend Mikrotik-Router zum Einsatz kommen, haben wir uns etwas näher mit der im RouterOS enthaltenen Funktion des Metarouters¹ beschäftigt: Dieser dient eigentlich primär dazu, dass WISP ihren Kunden einen „virtuellen“ Mikrotik-Router zur Verfügung stellen können, ohne sie auf das Hauptsystem zugreifen lassen zu müssen. Hierzu wird eine eigene Instanz aufgesetzt, eben ein sogenannter **Metarouter**.

Nur manche Routerboards **unterstützen** Metarouter, so auch der von uns eingesetzte Zentralrouter CRS125, aber auch die Grooves und Netmetals. Lt. Mikrotik unterstützen folgende Devices Metarouter: RB400, RB700 Serie ohne SPI Flash, SPI Flash, RB900 Serie ohne SPI Flash, RB2011 Boards. Weiters einige Routerboards mit PPC-Architektur, diese habe wir aber nicht für unser Projekt zum Laufen gebracht (getestet an einem RB800).

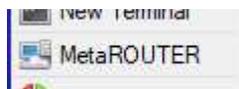
Aber was nützt uns ein zweiter virtueller Mikrotik-Router? Der kann ja auch nicht mehr wie der erste? Richtig, aber der Clou des Metarouters ist, dass auch ein OpenWRT-Image zur Verfügung steht und dieses in weiterer Folge um einen Apache-Webserver erweitert werden kann!

OpenWRT-Image

Wir laden uns vorerst das OpenWRT-Image von <http://openwrt.wk.cz/trunk/mr-mips/openwrt-mr-mips-rootfs-31411-basic.tar.gz> und speichern es auf unserem Rechner. Anschließend laden wir es via Winbox oder FTP in unser (**normales**, nicht virtuelles) Routerboard unter Files. Direkt in unser Routerboard bekommen wir es ansonsten auch mit `"/tool fetch url=http://openwrt.wk.cz/trunk/mr-mips/openwrt-mr-mips-rootfs-31411-basic.tar.gz".`

Anlegen des Metarouters

Wir suchen nun in der Winbox in der Menüleiste den Punkt



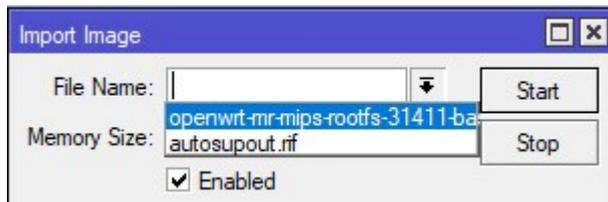
Ist dieser nicht da, unterstützt der verwendete Mikrotik-Router leider keinen Metarouter...

Im erscheinenden Metarouter-Menü klicken wir nun auf „Import Image“:

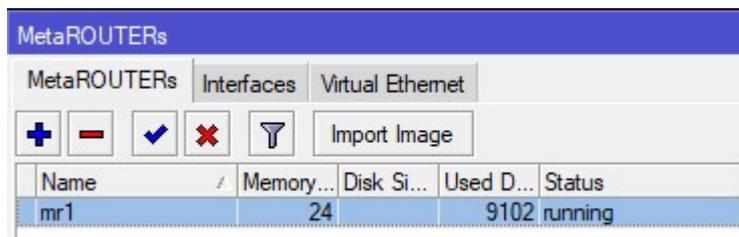


¹ <https://wiki.mikrotik.com/wiki/Manual:Metarouter>

Ein Auswahlfenster öffnet sich und wir wählen nun das zuvor auf unser Routerboard geladene OpenWRT-Image aus. Als „Memory size“ empfehlen sich (wenn sie am eigentlichen Router nicht zu sehr abgehen) 24 MB (sonst 16²) und dann klicken wir auf „Start“

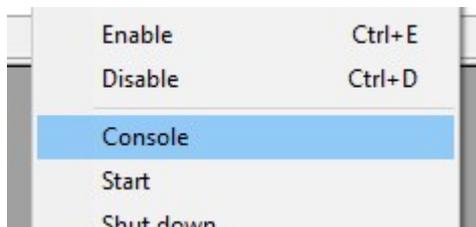


Das Image wird nun importiert und unten läuft eine Prozentanzeige durch. Nach Fertigstellung des Importvorganges steht uns nun der virtuelle Router zur Verfügung:



Zugriff auf den Metarouter

Unser virtuelles OpenWRT-Router ist nun da – aber wie greifen wir auf ihn zu? Dazu klicken wir mit rechts auf den angelegten Metarouter und wählen „console“:

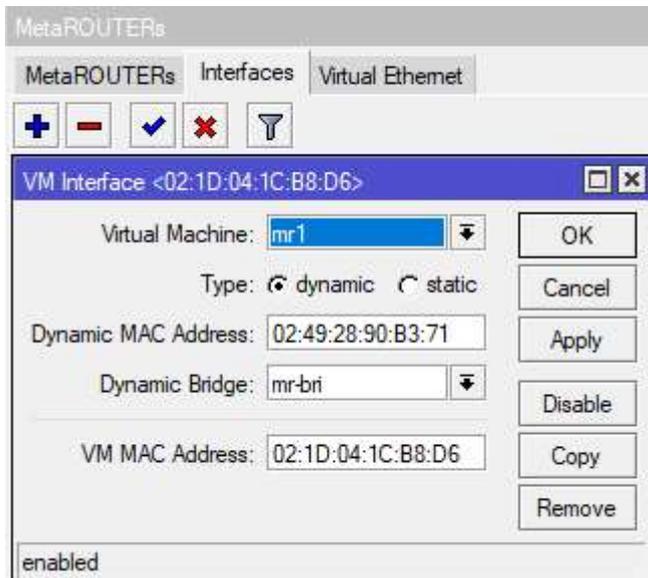


Es öffnet sich ein Terminalfenster und nach ein paar Enter-Eingaben sind wir an der Eingabeaufforderung:

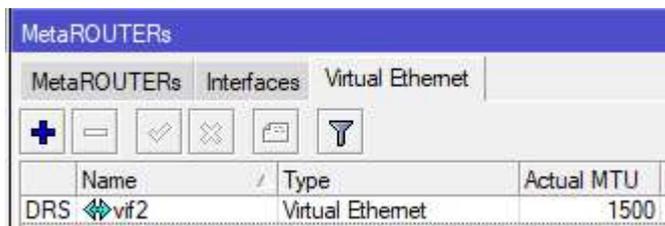
² Aus dem Mikrotik-Handbuch: „It is possible to run other virtual machines with less than 24MB RAM per machine if the virtual operating system is OpenWRT. The 24MB limitation is only for virtual RouterOS installations.“

Wir gehen nun nach Metarouter -> Interfaces und legen mit + ein neues Interface an.

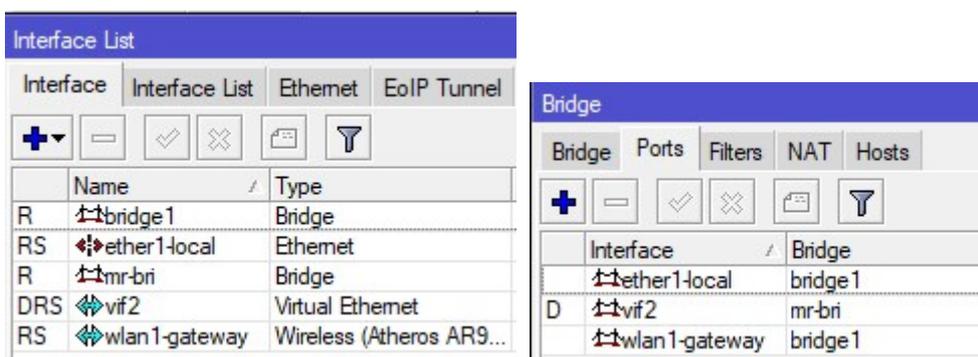
Dazu wählen wir die zuvor angelegte virtuelle Maschine „mr1“ aus, und binden diese dynamisch an die zuvor angelegte Bridge „mr-bri“. Die MAC-Adressen werden von ROS generiert:



Wenn wir nun einen Reiter weiter schauen zu „virtual ethernet“, sehen wir ein damit automatisch angelegtes virtuelles Interface vif:



Dieses findet sich auch in unserer Interface-Liste und (dynamisch) in der Bridge wieder:



Wir starten nun wieder die Konsole des Metarouters, da wir im Metarouter noch die Netzwerk-Konfiguration anpassen müssen.



Mit „vi /etc/config/network“ öffnen wir die entsprechende Datei.

Zu vi: Dieser ist etwas anders zu bedienen als übliche Editoren – nach dem Start ist man in Betrachter-Modus, kann aber nichts verändern! Wir drücken daher „i“ und können nun editieren und passen die Konfigurationsdatei wie folgt an (IPs natürlich an das eigene Netzwerk anpassen):

```
MetaROUTER mr1
# Copyright (C) 2006 OpenWrt.org

config interface loopback
    option ifname    lo
    option proto     static
    option ipaddr    127.0.0.1
    option netmask   255.0.0.0

config interface lan
    option ifname    eth0
    option type      bridge
    option proto     static
    option ipaddr    192.168.80.2
    option netmask   255.255.255.0
    option gateway   192.168.80.1
    option dns       192.168.80.1
```

Wir vergeben also der Bridge im Metarouter eine statische IP, hier die Bridge des normalen Routers (mr-bri 192.168.80.1) um eines erhöht, also *192.168.80.2* – weiters definieren wir auch noch **Netzmaske**, **Gateway** und **DNS** (die beiden letzteren spielt im Regelfall wohl wieder unsere mr-bri *192.168.80.1*).

Und wie verlassen wir vi nun wieder bzw. speichern unsere Änderungen ab? Zuerst drücken wir die ESC-Taste und anschließend geben wir „:wq“ ein und bestätigen mit Enter. Erledigt 😊

Als nächstes bearbeiten wir mit „vi /etc/resolv.conf“ den DNS-Eintrag und passen ihn an (das mit „i“, ESC und „:wq“ ist immer noch gleich):

```
MetaROUTER mr1
search lan
nameserver 192.168.80.1
~
~
```

Anschließend Neustart des Netzwerks mit /etc/init.d/network restart

```
MetaROUTER mr1
root@metarouter:/# /etc/init.d/network restart
[ 0.000000] br-lan: port 1(eth0) entering disabled state
[ 0.000000] device eth0 left promiscuous mode
[ 0.000000] br-lan: port 1(eth0) entering disabled state
[ 0.000000] device eth0 entered promiscuous mode
[ 0.000000] br-lan: port 1(eth0) entering forwarding state
root@metarouter:/#
```

Nun testen wir mal die Verbindung ins Internetz (bzw. Hamnet):

```
root@metarouter:/# ping orf.at
PING orf.at (194.232.104.141): 56 data bytes
64 bytes from 194.232.104.141: seq=0 ttl=55 time=21.128 ms
64 bytes from 194.232.104.141: seq=1 ttl=55 time=22.953 ms
64 bytes from 194.232.104.141: seq=2 ttl=55 time=19.558 ms
^C
--- orf.at ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 19.558/21.213/22.953 ms
root@metarouter:/#
```

Erfolgreich ☺

Raus kommen wir also – rein habe ich zb mit einer dst-nat-Regel gelöst, die einen Port des normalen Routers auf die 192.168.80.2:80 umbiegt. Aber noch läuft ja noch gar kein 80er-Port in unserem Metarouter.

Installation apache-Webserver

„vi /etc/opkg.conf“ – dort passen wir die erste Zeile an:

```
src/gz snapshots http://openwrt.wk.cz/trunk/mr-mips/packages
```

(in dem repository gibt's auch den editor nano, wer mit vi nicht so klar kommt)

Dann lesen wir die Paketliste neu ein und installieren Apache:

```
opkg update
```

```
opkg install apache
```

Anschließend booten wir den „normalen“ Router neu und starten anschließend in der Metarouter-Konsole Apache:

```
apachectl start
```

Fertig – falls der Port forward noch nicht erledigt ist oder man das Routing anders lösen will in den Metarouter rein, nun durchführen und anschließend die entsprechende IP aufrufen. Es begrüßt uns der Apache-default-Text:



It really works!

Die Webseiten können wir nun in /usr/share/htdocs bearbeiten (mit vi, viel Spaß ;-)).

Noch zwei Hinweise:

Das Passwort des Metarouters kann man mit „passwd“ setzen.

Und falls eine genaue Uhrzeit nötig ist:

```
opkg update
opkg install ntpd
/etc/init.d/sysntpd disable
/etc/init.d/ntpd enable
/etc/init.d/ntpd start
```

Den Zeitserver kann man in /etc/config/system bearbeiten.

Bernd, OE7BSH oe7bsh@firac.at

v 1.0 – 20.8.2017

Quellen:

<https://aacable.wordpress.com/2015/02/16/expanding-possibilities-howto-add-3rd-party-packages-in-mikrotik-kvmmetarouter/>

<http://gregsowell.com/?p=2679>

<https://wiki.mikrotik.com/wiki/Manual:Metarouter>