Ergänzung zum Beitrag in FA 2/18, S. 188 f. „C4FM-Fusion-Repeater mit WIRES-X-Anbindung via LTE“

Da im o. g. Beitrag aus Platzgründen die Eingaben zur Konfiguration von Raspberry Pi, Fritzbox und WIRES-X-Rechner entfallen mussten, sind diese hier aufgelistet. Die Weblinks [3] bis [7] sind in der gedruckten Ausgabe auf S. 189 zu finden.

■ Beispieladressen

Exemplarisch werden hier folgende Einstellungen (z. B. DynDNS-Adresse) verwendet, die vom Nutzer anzupassen sind:
– DynDNS-Adresse der Fritzbox: *a1b2c3d4e5f6g7h8.myfritz.net*
– IP-Adresse der Fritzbox (Gateway, DNS-Server): *192.168.178.1*
– IP-Adresse des Raspberry Pi 3: *192.168.178.50*
– VPN-IP-Adresse des Open-VPN-Servers: *10.168.51.1*
– VPN-IP-Adresse des Open-VPN-Clients: *10.168.51.6*

■ Konfiguration des Raspberry Pi

Alle Befehle (kursiv) werden auf der Konsole eingegeben:

1. Aktualisieren des Raspberry

sudo apt-get update && sudo apt-get upgrade

2. Festlegen der statischen IP-Adresse:

sudo nano /etc/network/interfaces

Sollte die Zeile *iface eth0 inet manual* vorhanden sein, muss diese auskommentiert werden.

#iface eth0 inet manual

Die Konfiguration für die statische IP-Adresse sieht in unserem Beispiel wie folgt aus:

```
# IPv4 address
iface eth0 inet static
        address 192.168.178.50
        netmask 255.255.255.0
        network 192.168.178.0
        broadcast 192.168.178.255
        gateway 192.168.178.1
```

Nach den Änderungen muss der Netzwerkdienst einmal durchgestartet werden:

sudo systemctl restart networking.service

Jetzt sollte die neue Konfiguration bereits angewendet werden. Dies kann einfach mit *ifconfig* überprüft werden.

3. Installation von OpenVPN:

sudo apt-get install openvpn

4. Erzeugen der Open-VPN-Zertifikate für den Server (*KEY_NAME=server*) und die WIRES-X-Node (*KEY_NAME=WIRES-X-Node*), wie in [3] für Windows beschrieben:
Folgende Zertifikate gehören in das Verzeichnis */etc/openvpn/easy-rsa/keys*:
– *ca.crt*
– *server.crt*
– *server.key*
– *tlsauth.key*
– *dh4096.pem*
Aus Sicherheitsgründen werden die Berechtigungen für die Zertifikate neu gesetzt:

```
sudo chown -R root:root /etc/openvpn/
  easy-rsa/keys/
sudo find /etc/openvpn/easy-rsa/keys/ -type f
  -exec chmod 400 {} \;
```

Jetzt kann die Konfigurationsdatei für den Server erzeugt werden (siehe Listing 1). Diese liegt unter */etc/openvpn/server.conf*:

sudo nano /etc/openvpn/server.conf

Um die Konfiguration abzusichern, wird die Berechtigung neu gesetzt:

sudo chmod 600 /etc/openvpn/server.conf

Der Open-VPN-Server kann jetzt gestartet werden:

```
sudo systemctl enable openvpn.service
sudo systemctl start openvpn.service
```

Ab jetzt sollte unter *ifconfig* auch ein *tun0*-Device auftauchen.
5. IPv4 Forwarding aktivieren: Dazu muss in der Datei */etc/sysctl.conf* die Zeile *net.ipv4.ip_forward=1* aktiviert werden. Damit die Änderung übernommen wird, muss die Konfiguration gespeichert werden:

sudo sysctl –p

6. Einstellen der Firewall und des UDP-Routings: Damit auch nach einem Neustart des Raspberry alles noch funktioniert, wird das Paket *iptables-persistent* installiert:

sudo apt-get install iptables-persistent

Listing 1: server.conf

```
dev tun
proto udp
port 1194
status /var/log/openvpn-status.log
log /var/log/openvpn.log
verb 3
persist-tun
persist-key
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key
dh /etc/openvpn/easy-rsa/keys/dh4096.pem
tls-auth /etc/openvpn/easy-rsa/keys/tlsauth.key 0
cipher AES-256-CBC
auth SHA512
keepalive 10 120
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC
-SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC
-SHA256:TLS-DHE-RSA-WITH-AES-256-CBC-SHA
server 10.168.51.0 255.255.255.0
tls-server
tls-version-min 1.2
auth-nocache
# allen Client Traffic nur durch den VPN Tunnel schicken
push "redirect-gateway def1"
push "dhcp-option DNS 192.168.178.1"
push "block-outside-dns"
```

Die *iptables*-Regeln werden in einem kleinen Shell-Skript gespeichert, das bei Anpassungen schnell wieder ausgeführt werden kann (siehe Listing 2).

sudo nano /usr/local/bin/iptables.sh

Das Skript wird zuerst als ausführbar markiert und dann direkt gestartet:

sudo chmod +x /usr/local/bin/iptables.sh &&
sudo sh /usr/local/bin/iptables.sh

Die geschriebenen Regeln können bei Bedarf nachträglich überprüft werden:

sudo nano /etc/iptables/rules.v4

Damit ist der Open-VPN-Server einsatzbereit. Der gesamte Internetverkehr des WIRES-X Rechners läuft jetzt durch den VPN-Tunnel, und die sechs UDP-Ports werden vom DSL-Anschluss über das Mobilfunknetz zum WIRES-X-Rechner geroutet.
7. Wenn man Bluetooth und WLAN auf dem Raspberry Pi nicht benötigt, kann man es auch abschalten. Dazu wird z. B. das Laden der Treiber mit einer Konfigurationsdatei verhindert und der entsprechende Dienst abgeschaltet [4] (raspi-blacklist .conf).

■ Konfiguration der Fritzbox

1. Die Portweiterleitungen für Open VPN (1194 UDP) und die sechs UDP-Ports für den WIRES-X-Link (46100, 46110, 46112, 46114, 46120 und 46122) werden, wie in [5] beschrieben, eingerichtet.
2. Nach der Aktivierung der Weiterleitungen, kann der Raspberry Pi die Verbindungen aus dem Internet auf diesen Ports unter der DynDNS-Adresse der FritzBox entgegennehmen.

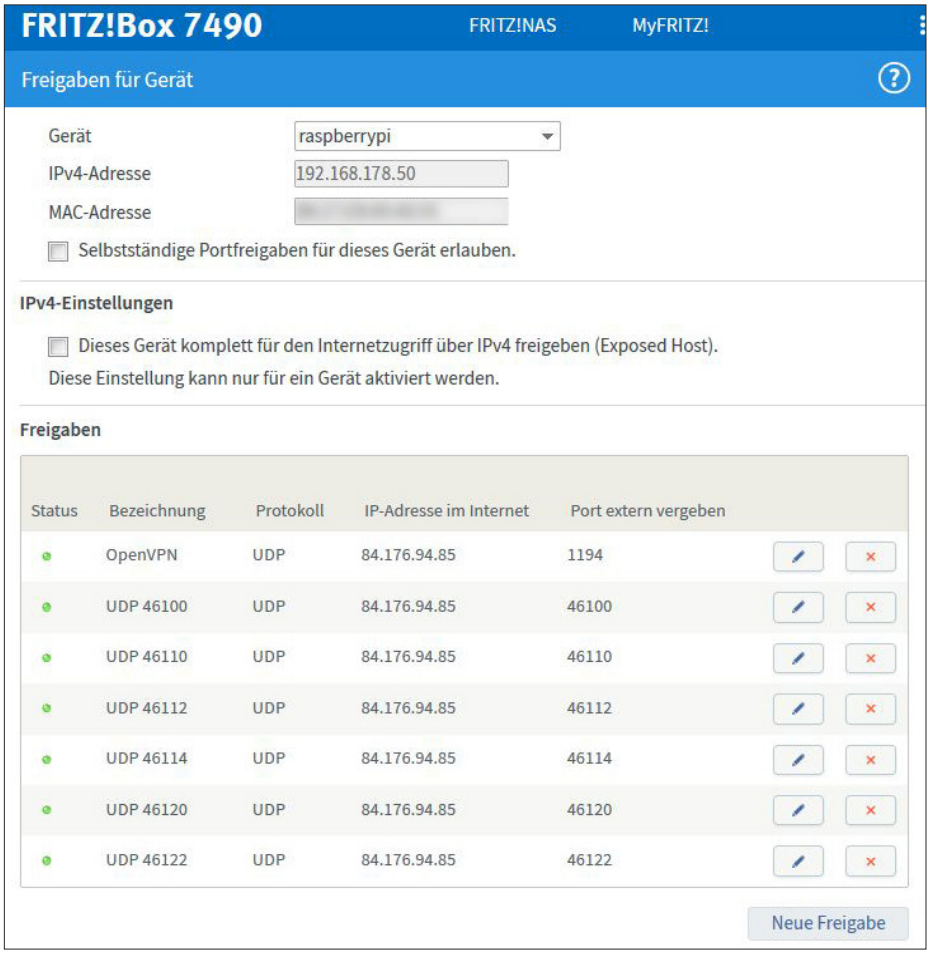

Bild 3: Portforwarding auf der Fritzbox

■ Konfiguration des WIRES-X-Rechners (Windows)

1. Für die Installation der OpenVPN-Software lädt man sich den Open-VPN-Installer für Windows von *https://openvpn.net/index.php/open-source/downloads.html* herunter und führt die Datei als Administrator aus.
2. Die folgenden Zertifikate, die wir bereits bei der Konfiguration des Raspberry Pi erzeugt haben, werden in das Verzeichnis *C:\Programme\OpenVPN\config* kopiert:

Listing 2: iptables.sh

```
#!/bin/bash
iptables -t filter -F
iptables -t nat -F
iptables -t nat -A PREROUTING -i eth0 -p udp -m udp --dport 46100 -j DNAT --to-destination 10.168.51.6:46100
iptables -t nat -A PREROUTING -i eth0 -p udp -m udp --dport 46110 -j DNAT --to-destination 10.168.51.6:46110
iptables -t nat -A PREROUTING -i eth0 -p udp -m udp --dport 46112 -j DNAT --to-destination 10.168.51.6:46112
iptables -t nat -A PREROUTING -i eth0 -p udp -m udp --dport 46114 -j DNAT --to-destination 10.168.51.6:46114
iptables -t nat -A PREROUTING -i eth0 -p udp -m udp --dport 46120 -j DNAT --to-destination 10.168.51.6:46120
iptables -t nat -A PREROUTING -i eth0 -p udp -m udp --dport 46122 -j DNAT --to-destination 10.168.51.6:46122
iptables -t nat -A POSTROUTING -s '10.168.51.0/24' -j MASQUERADE
iptables -A FORWARD -d '10.168.51.6/32' -p udp -m udp --dport 46100 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d '10.168.51.6/32' -p udp -m udp --dport 46110 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d '10.168.51.6/32' -p udp -m udp --dport 46112 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d '10.168.51.6/32' -p udp -m udp --dport 46114 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d '10.168.51.6/32' -p udp -m udp --dport 46120 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d '10.168.51.6/32' -p udp -m udp --dport 46122 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT
iptables -A FORWARD -j REJECT
iptables-save > /etc/iptables/rules.v4
```

– *ca.crt*
– *WIRES-X-Node.crt*
– *WIRES-X-Node.key*
– tlsauth.key

3. Jetzt wird im gleichen Verzeichnis die Konfigurationsdatei *WIRES-X-Node.ovpn* für den Client mit einem Texteditor erstellt (siehe Listing 3).

4. Die Autostartfunktion von Open-VPN sollte aktiviert werden, sodass sich der Rechner beim Systemstart automatisch mit dem Open-VPN-Server verbindet [6].

5. Steht der Open-VPN-Tunnel und ist alles richtig konfiguriert, zeigt der *Port Check* der WIRES-X-Software für alle sechs UDP-Ports jetzt den Status *OK* an.

Nach der erfolgreichen Registrierung bei Yaesu ist die Node unter ihrer ID im WIRES-X-System über das Internet erreichbar [7].

Listing 3: WIRES-X-Node.ovpn

```
client
dev tun
proto udp
remote a1b2c3d4e5f6g7h8.myfritz.net 1194
resolv-retry infinite
nobind
persist-key
persist-tun
float
remote-cert-tls server
verb 3
cipher AES-256-CBC
auth SHA512
mute-replay-warnings
ca ca.crt
cert WIRES-X-Node.crt
key WIRES-X-Node.key
tls-auth tlsauth.key 1
```