26.02.2018

Zugriff auf das Heimnetz und Nutzung eines (eigenen) VPN mit Macbook (Air), iPhone & Co. an einem DS-Lite-Anschluss mit Hilfe einer Fritzbox, des MyFritz!-Dienstes, einem Raspberri Pi als FipBox und einem universellen Portmapper von www.feste-ip.net:

Soviel vorab: Es funktioniert super! Die (einmalige) Einrichtungsarbeit lohnt sich meiner Meinung nach auf jeden Fall.

Wie viele andere bin ich an einem DS-Lite - Anschluss (von Unitymedia). Als Router fungiert eine Fritzbox 6360 Cable. Geräte im Heimnetz können von außen nicht erreicht werden (jedenfalls soweit sie kein IPV6 unterstützen und/oder weil man nicht überall über einen IPV6-Anschluss verfügt [wie das bei mir z.B. mobil im O2-Netz der Fall ist...].

DS-Lite bedeutet (u.a.), dass der an diesem Anschluss werkelnde Router keine (eingehende) IPV4-Adresse hat (auch nicht dynamisch!). Vielmehr besteht eine Erreichbarkeit von außen (nur) mittels IPV6. Um die sich daraus ergebenden Auswirkungen zu verstehen, ist in nachfolgender Tabelle angegeben, wie man Geräte im Heimnetz (am Beispiel Webcam) erreichen kann, je nachdem wie sie verbunden sind bzw. von wo erreicht werden sollen (Firewalls bzw. Freigaben lassen wir hier mal weg):

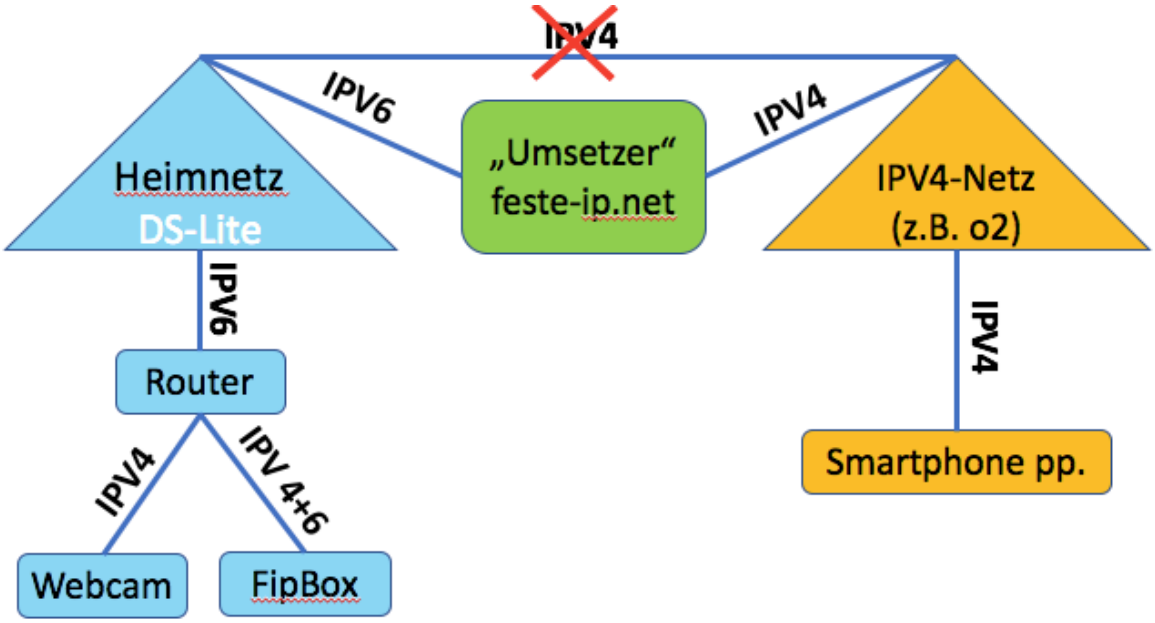
	Anschluss Cam:		**Internetanschluss:**		**Erreichbarkeit über:**	
	IPV4	**IPV6**	**IPV4**	**IPV6 (DS-Lite)**	**IPV4**	**IPV6**
Webcam 1	X		X		X	--
Webcam 2		X		X	--	X
Webcam 3	X			X	--	--

Aus der Tabelle kann man „das Problem“ erkennen. Während die Webcam 1 (im Heimnetz verbunden mit IPV4 an einem Internetanschluss über IPV4) über einen IPV4-Anschluss von außen (problemlos) erreichbar ist, sieht dies bei Webcam 2 schon anders aus. Sie ist über IPV6 im Heimnetz verbunden und der Internetanschluss verfügt nur über eine von außen erreichbare IPV6-Adresse. Also können wir das Gerät auch nur über ein IPV6-Netz von außen erreichen. Ganz blöde wird es, wenn wir ein Gerät (wie üblich!) mit IPV4 im Heimnetz verbunden haben, der Internetanschluss aber nur über IPV6 verfügt (wie nun mal bei DS-Lite). Bei einem Versuch, die Webcam 3 (von außen) zu erreichen, scheitern wir gnadenlos. Und genau diesen Umstand wollen wir nun lösen, wobei wir zugleich auch eine Lösung für „Webcam 2“ erhalten.

Schon länger nutzte ich - um zumindest den Router fernwarten zu können - den MyFritz-Dienst in Kombination mit einem universellen Portmapper von www.feste-ip.net. Der Portmapper wurde also von einem kommerziellen Service angeboten. Der Service ist günstig (auf das Jahr bezogen max. 4,95 €; weitere Details unten) und schnell eingerichtet und vor allem: es klappte auch immer prima. Dabei wollte ich also bleiben.

Jetzt wollte ich aber die „größere Lösung“, nämlich neben der Erreichbarkeit aller (gewünschten) Geräte im Heimnetz auch die Einrichtung eines (eigenen, privaten) VPN-Tunnels. Damit kann man sich auch mit einem (noch so weit) entfernten Gerät (also von außen, auch mobil) in das eigene Netz einbinden. Praktisch ist das dann so, als hätte man es mit einem Netzwerkkabel bzw. WLAN mit dem Heimnetz verbunden. Man kann dann also direkt (ohne weitere Portmapper pp.) - und verschlüsselt! - per FTP, SSH, VNC bzw. auf NAS oder auf Webcams pp. zugreifen und auch im Internet surfen. Natürlich soll das Ganze auch von meinem Smartphone (iPhone) aus funktionieren...

Folgendes Bild veranschaulicht nochmals die (herzustellende) Konfiguration, damit das alles (trotz bzw. mit DS-Lite) läuft:



Apropos VPN (Virtual Private Network): Damit erreicht man einen sicheren Datenaustausch im Internet. Die Daten laufen dann nämlich verschlüsselt über einen „Tunnel“, so dass Dritte unbefugt nichts abfangen können. Besonders in öffentlichen Netzwerken wie Internetcafés und Hotels haben Hacker leichtes Spiel, Anmeldedaten und Passwörter abzufangen. Alle, die sicher und anonym im Internet surfen möchten, sichern ihre Verbindung mit einem verschlüsselten VPN-Tunnel ab (bzw. sollten das). Praktisch wird dies insbes. von Firmen eingesetzt, damit Mitarbeiter von Zuhause aus Zugriff auf Rechner im Firmennetz erhalten. Geht man dann über den Tunnel ins Internet, ist dies genauso sicher bzw. unsicher, wie wenn man das von Zuhause macht.

Ziel also klar (ich haben will!); **nun zur Umsetzung:**

Auf der Seite von feste-ip.net (http://www.feste-ip.net/fip-box/allgemeine-informationen/) wird eine sog. FipBox beschrieben, die man dort auch fertig kaufen kann. Dort wird auch die Installation einer solchen Box (mit zur Verfügung gestellten Scripten) beschrieben. Ich habe mich dann dafür entschieden, die Box selbst zu installieren und einzurichten. Beschafft habe ich mir dazu einen Raspberry Pi 3 mit 16GB SD-Karte, Netzteil, Gehäuse und Kühlrippen für die IC´s. Für die Umsetzung wählte ich bei mir die Hardware Macbook Air und ein iPhone 7 (mit beiden bin ich häufig mobil unterwegs).

Bei der Installation/Einrichtung der Software ging (leider und warum auch immer selbst mit dem Script von feste-ip.net) eine ganze Menge schief - aber jetzt funktioniert die Kiste wunderbar! Die „Fehler“, die mich viele Stunden meines Lebens gekostet aber im Rahmen der Beseitigung auch mal wieder ein ganzes Stück schlauer gemacht haben, möchte ich der Nachwelt ersparen. Daher hier das aus meiner Sicht Wichtigste - so wie es bei mir jedenfalls funktioniert:

A. Die Einrichtung der FipBox, die Erstellung notwendiger Freigaben an der FritzBox und Einrichtung eines universellen Portmappers bei feste-ip.net:

1. Die Speicherkarte für den Raspberry Pi (es geht auch Modell 2) mit dem Image Raspian Jessie beschreiben (ich habe mich für „RASPBIAN STRETCH WITH DESKTOP in der Version von November 2017 mit Kernel-Version 4.9“ entschieden). Wo es das Image gibt und wie das Beschreiben der SD-Karte funktioniert (wofür ich am Mac „ApplePi-Baker“ verwendet habe), steht z.B. hier: http://www.feste-ip.net/fip-box/basic/fip-box-installieren/

2. Die nun frisch beschriebene SD-Karte NOCH NICHT direkt in den Raspberry Pi einlegen. Die neueren Images haben nämlich als Standard den SSH-Zugriff deaktiviert, d.h. nach Start des Pi hat man darauf - soweit man nicht Bildschirm und Tastatur anschließt, ihn also „headless“ betreibt - leider keinen Zugriff. Um den Start mit SSH zu ermöglichen, muss man auf der SD-Karte im Boot-Verzeichnis noch eine **LEERE** (Text-)Datei mit dem Namen „ssh“ anlegen (kann mit einem beliebigen Texteditor erledigen). Erst nachdem dies erfolgt ist, sollte man die Karte in den Pi stecken und das Gerät (natürlich auch an den Router angeschlossen) starten.

3. In einem Terminalfenster (beim Mac oder Linux, ansonsten über Putty o.ä.) mit dem Pi auf der Kommando-Ebene verbinden mit dem Befehl „ssh pi@IP-Adresse“ (die IP-Adresse des Pi natürlich entsprechend einsetzen; ggfs. im Router nachsehen, welche IP das Gerät zugewiesen bekommen hat).

   Bitte beachten: Die lokale IPV4-Adresse der Box sollte vom Router fest (also immer gleich) vergeben werden. Dies können wir in der FritzBox unter Heimnetz - Heimnetzübersicht - in den Details zur Box einstellen (anhaken) und heißt dort: „Diesem Netzwerkgerät immer die gleiche IPv4-Adresse zuweisen“.

   Wenn nun der Befehl ssh... erfolgreich abgesetzt ist, wird man nun noch um Eingabe des Passworts gebeten. Dies lautet nach frischer Installation - ganz geheim“ - „raspberry“. Damit landen wir dann auf der (sehr mächtigen) Konsole!

4. Jetzt kann man dank des Scripts von fest-ip.net aus dem Pi eine FipBox machen. Dazu gibt man an der Konsole nacheinander diese Kommandos ein:

   - **sudo su**
   - **cd /tmp**
   - **wget www.portmapper.de/fipbox/raspian2fipbox.sh**
   - **sh raspian2fipbox.sh**

   Am Ende ist ein neues Passwort zu vergeben. Allerdings sollte man das auf der Seite von feste-ip.net danach angegebene Kommando „reboot“erst mal **NICHT**

eingegeben (sonst startet das Teil zwar schön neu, eine Verbindung ist aber - so war es jedenfalls bei mir - nicht mehr möglich! (genau genommen soweit man nicht Bildschirm und Tastatur anschließt, aber das Teil soll ja in unser Netz - und eben das tut es [tat es bei mir] dann eben nicht mehr).

5. Das Script raspian2fipbox.sh legt u.a. die Textdatei /etc/network/interfaces an bzw. modifiziert sie. Den Inhalt muss man (musste ich!) ändern, um das vorbeschriebene Problem zu lösen. Zum Ändern öffnet man die Datei mit einem Editor, z.B. mit „nano“ durch folgendes Kommando:

   sudo nano /etc/network/interfaces

   Der Inhalt der Datei sollte wie folgt aussehen (das was da mehr drin steht, ist zu löschen!).

   ```
   # interfaces(5) file used by ifup(8) and ifdown(8)

   # Please note that this file is written to be used with dhcpcd
   # For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

   # Include files from /etc/network/interfaces.d:
   source-directory /etc/network/interfaces.d

   auto eth0:1
   ```

   Anschließend speichern und verlassen mit CTRL-X sowie Bestätigung mit Y(es).

6. Nun (erst) kann/sollte das System neu gestartet werden. Dazu den Befehl eingeben:

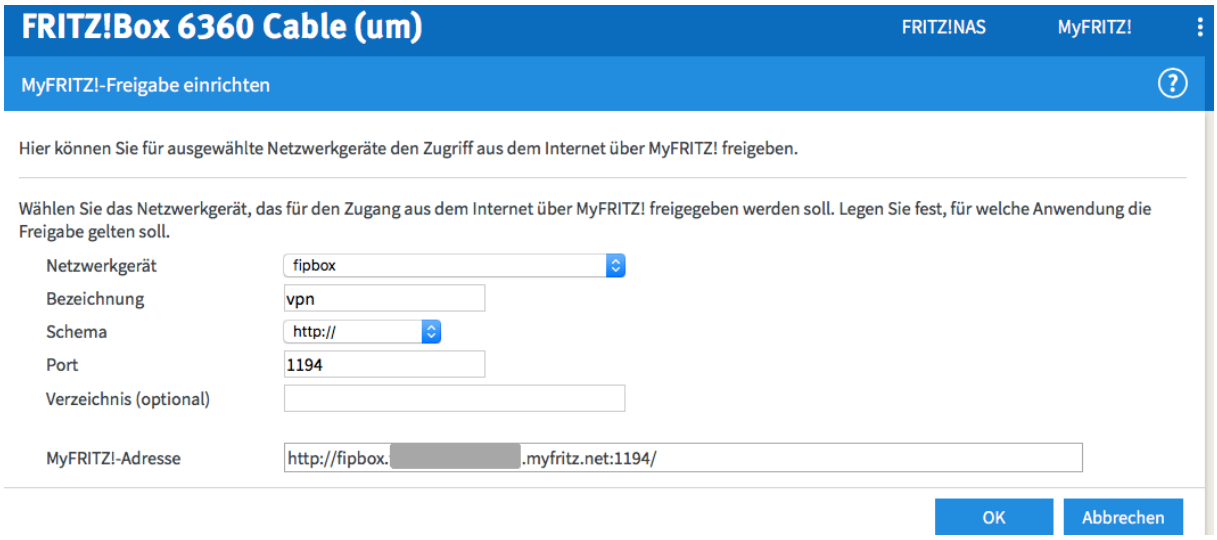
   sudo reboot

7. Wenn alles funktioniert hat, startet der Pi neu und ist nun unter dem Namen „fipbox“ im eigenen Netz erreichbar. Anmelden kann (und sollte!) man sich nun wieder über ein Terminal (siehe Nr. 3, nun mit dem neu vergebenen Passwort). Jetzt meldet sich „das Teil“ auch so und es steht (wie am Bildschirm angegeben) unter anderem der Befehl „fipedit“ zur Verfügung (den wir dann etwas später noch brauchen werden...).

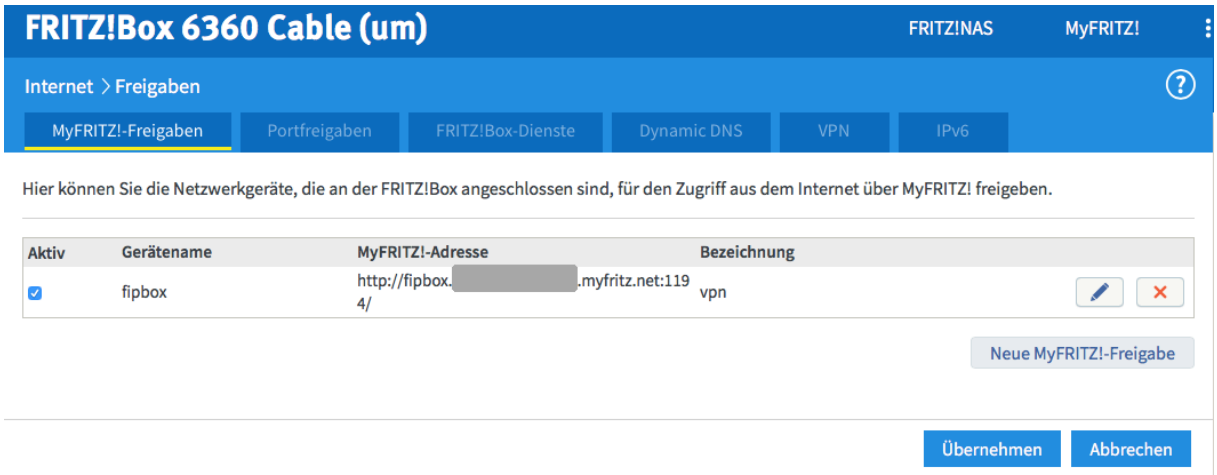
Erläuterung:
Um nun Geräte im „nur“ mit IPV6 erreichbaren Heimnetz von überall (auch mobil mit den dort teils nur verfügbaren IPV4-Adressen) erreichen zu können, bedarf es einer festen (statischen) IPV4-Adresse. Diese haben wir bei DS-Lite ja eben nicht. Man kann sich solche Adressen aber über universelle Portmapper bei feste-ip.net einrichten. Die Mapper müssen jedoch auch „wissen“, wo die Reise der Datenpakete hingehen sollen (logischerweise in unser Heimnetz auf die FipBox). Leider übermitteln die Geräte bei IPV6 (also auch bei DS-Lite) nicht die ganze IP-Adresse, sondern nur den externen Teil. Dieser Teil reicht aber nicht, um ein konkretes Gerät anzusprechen (es fehlt die Netzwerkpräfix). Um mit diesem Dilemma umzugehen, gibt es einen Dienst, den MyFRITZ! (jedenfalls für Fritzbox´en) kostenlos zur Verfügung stellt. Dort muss man sich - falls nicht schon geschehen - erst mal registrieren. Also

8. Soweit nicht bereits geschehen, ein MyFritz!-Konto anlegen. Dies geht unproblematisch über die Fritzbox (Routermenü) über Internet - MyFritz!-Konto.

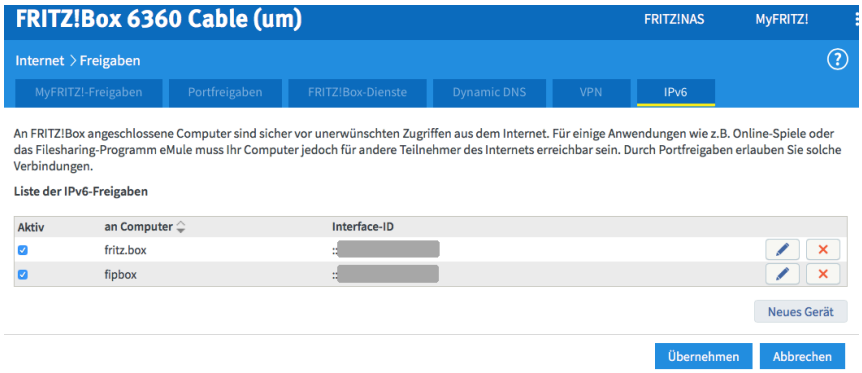
9. Damit der universelle Portmapper (später, hoffentlich sehr bald!) das von uns gewünschte Gerät (also die FipBox) erreichen kann, bedarf es auch entsprechender Freigaben im Router. Dies erledigen wir im Routermenü unter Internet - Freigaben - MyFritz-Freigaben und sollte dann so aussehen:



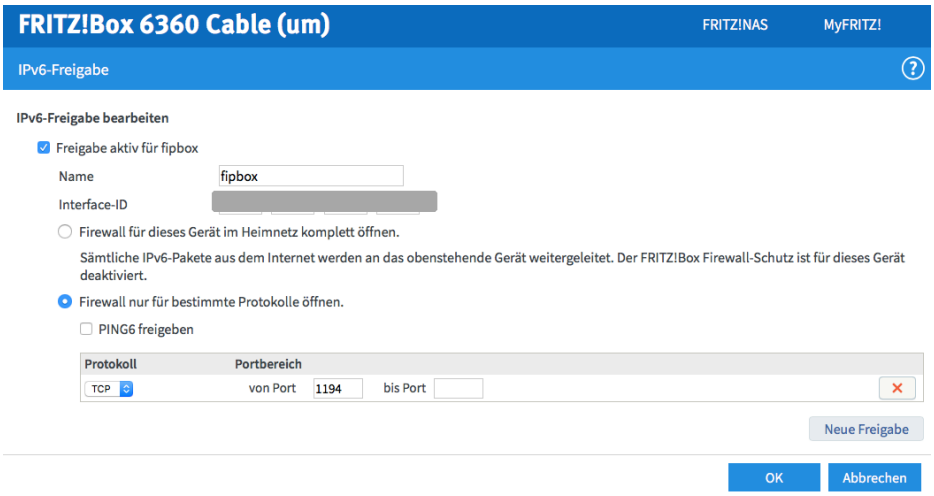
Die Eingaben sollten - wie oben zu sehen - erledigt werden. Die unten angegebene MyFritz!-Adresse wird (dann) automatisch vergeben. Diese brauchen wir dann für die Portmapper! (also notieren, kopieren pp.). Nach Klick auf OK sieht das Ganze dann so aus (auch hier ist die MyFritz!-Adresse zu sehen):



Automatisch erstellt wird dann von der Fritzbox freundlicherweise gleich auch noch der notwendige Eintrag bei Freigaben - IPV6. Das sieht dann so aus (wobei bei mir hier noch eine Freigabe für die FritzBox zu sehen ist, die ich noch von vorher hatte - diese braucht man aber nicht! und wurde von mir/bei mir inzwischen auch schon gelöscht):



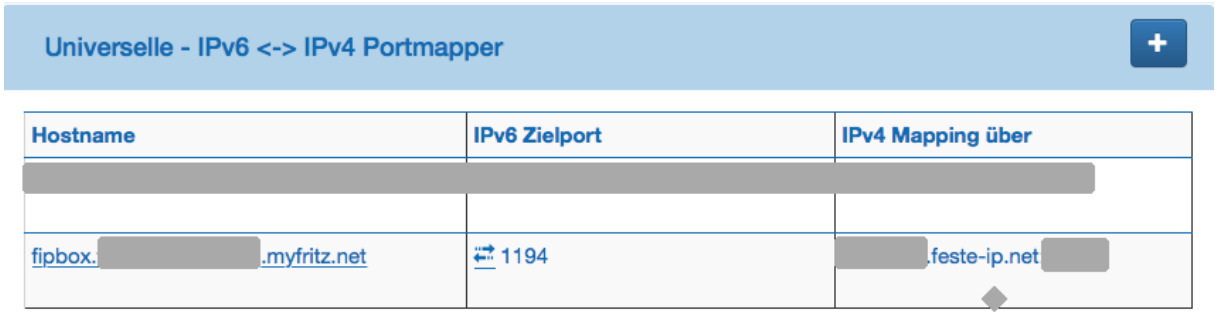
Durch Klicken auf den Bearbeitungsstift rechts (in der Zeile der „fipbox“) gelangt man zu folgender Ansicht (dort sollte die im Screenshot ausgegraute Interface-ID zu der FipBox passen; dies am besten auf der Konsole mit dem Befehl „ifconfig“ überprüfen):



Ok. Nun haben wir den „FritzBox-Teil“ erledigt und wenden uns dem universellen Portmapper zu.

10. Auf der Seite http://www.feste-ip.net - soweit nicht schon geschehen - registrieren und einen universellen Portmapper anlegen. Ja, das kostet nach einem freien Testzeitraum (von derzeit 50 Tagen) Geld. Der aufzuwendende Betrag fällt meiner Meinung nach jedoch sehr moderat aus. Der notwendige (eine) Portmapper kostet am Tag einen Credit (darüber könnte man insgesamt 12 Ports ansteuern und dann über die FipBox sofort auf Endgeräte im Netz leiten). Die nötigen Credits kauft man im Block. Je nach Größe des Einkaufs an Credits (von 365 - 10.000) beträgt der Preis nach derzeitigem Stand [21.02.2018] für einen Credit maximal 1,4 Cent (gerundet) bis minimal weniger als einen Cent (konkret 0,795 ct). Auf das Jahr bezogen - bei der teuersten (praktisch jährlichen) Einkaufsvariante - nur 4,95 €. Da kann man nun wirklich nicht meckern. Näheres findet sich - jeweils aktuell - hier: http://www.feste-ip.net/kosten/preise/

    Kommen wir zu den Einstellungen für den universellen Portmapper:



Wie man (hoffentlich) erkennen kann, ist nun als Hostname (also des gewünschten Ziels des Portmappers) die in Ziff. 9 notierte MyFritz!-Adresse anzugeben; und zwar mit dem (im Router ja freigegebenen) Port 1194. Dann gibt feste-ip.net die IPV4-Mapping-Adresse (rechts) aus, die wir später brauchen, um von außen auf unser Heimnetz (bzw. die FipBox, das VPN) zugreifen zu können (also unbedingt auch notieren, kopieren pp.).

Jetzt geht es noch daran, auf dem Pi bzw. der FipBox einen VPN-Server zu installieren und einzurichten.

11. Für die Einrichtung des VPN-Servers (konkret openvpn) stellt feste-ip.net ein Script zur Verfügung, dass nun ausgeführt werden muss. Das geht an der Konsole wie folgt (aber nicht wundern: Die mit dem letzten Befehl initiierte Ausführung des Scripts dauert leider recht lange...da muss man durch...):

    - **sudo su**
    - **cd /tmp**
    - **wget www.portmapper.de/fipbox/fipboxvpn.sh**
    - **bash fipboxvpn.sh**

12. Nachdem das Script dann (endlich) fertig ist, wird auf dem Bildschirm (Terminal) ein Link und - was noch wichtiger ist! - der Private Schlüssel angezeigt, den wir später unbedingt für die Einrichtung des VPN´s an den Endgeräten brauchen. Wir kopieren uns also den Teil von „BEGIN“ bis „END“ uns speichern diesen zunächst mal sicher irgendwo ab. Dabei sind die Zeilen „-----BEGIN PRIVATE KEY-----„ und „-----END PRIVATE KEY-----„ mit zu kopieren!

    Mit dem zusätzlich (in der Konsolenansicht) aufgeführten Link kann (und sollte) man vier notwendige Dateien (konkret: „ca.crt“, „client1.crt”, “client1.key” und “fipboxathome.ovpn”) auf ein Endgerät der Wahl (bei mir war es mein Macbook) herunterladen. Dazu such man sich am besten zuvor ein eigenes leeres Verzeichnis am Zielrechner aus bzw. legt ein solches neu an. Die vier Daten befinden sich danach dort. Sollte man dies unterlassen/vergessen, braucht keine Panik entstehen. Man kann das später auch noch anders machen (siehe unten unter B. 1.)

13. Leider ist es (bzw. war es bei mir) damit (abgesehen von den noch folgenden Punkten) noch nicht getan - oder kurz: es funktionierte nicht! Die Datei server.conf musste ich erst noch ändern. Dies geht wieder mit dem Editor mit folgendem Befehl:

    sudo nano /etc/openvpn/server.conf

Der Inhalt sollte dann so aussehen, wobei IP-Adresse in der mit „server“ beginnenden Zeile durch den Beginn des im Heimnetz verwendeten Adressraums zu ersetzen ist (also eine 0 am Ende haben muss, z.B. 192.168.1.0):

```
port 1194
proto tcp6-server
dev tun
cipher AES-256-CBC
dh /etc/openvpn/easy-rsa/keys/dh2048.pem
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key
server Beginn-IP-Adressauam 255.255.255.0
comp-lzo
persist-key
persist-tun
status /var/log/openvpn-status.log
log-append /var/log/openvpn/openvpn.log
verb 3
client-to-client
push "redirect-gateway def1 bypass-dhcp"
#set the dns servers
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
duplicate-cn
keepalive 10 120
user nobody
group nogroup
```

Anschließend wieder speichern und verlassen mit CTRL-X sowie Bestätigung mit Y(es).

Dann: Neustarten der FipBox mit:

sudo reboot

Nach dem Neustart der Box darauf wieder mit dem Terminal zugreifen und dann eingeben:

fipedit

Hier werden - automatisch wieder mit dem Editor nano - die aktiven Portmapper konfiguriert. Der Inhalt der Datei sollte nach Anpassung so aussehen:

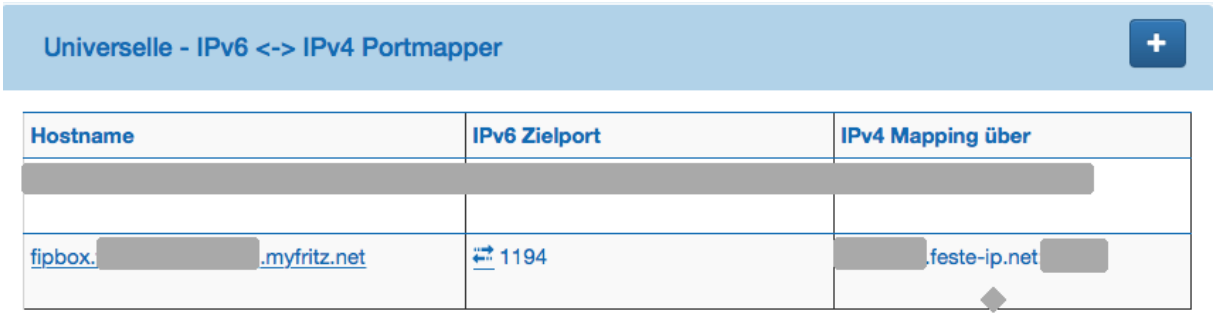
```
#INAKTIV#VPNMAPTCP#80#192.168.12.3#80
#INAKTIV#VPNMAPUDP#53#192.168.12.3#53

#INAKTIV#DNS#ihr-name.feste-ip.net#HOSTID#Passwort
#INAKTIV#PORTMAPPER#IPv6Port#IPv4ZielIP#IPv4Port
#AKTIV#PORTMAPPER#1194#192.168.X.XX#1194
```

Interessant ist hier nur die letzte Zeile! Sie gibt der FipBox an, was sie mit der/einer eingehenden Verbindung machen soll, nämlich hier: Den eingehenden Port 1194 weiterleiten an die IP 192.168.1.11 (des Heimnetzes) als Port 1194. Sie haben es sicher gewusst/geahnt, die genannte IP muss die der FipBox (im Heimnetz) sein, ist also entsprechend anzupassen. Die anderen Zeilen kann man so lassen, wie sie sind (jedenfalls sollten alle anderen Mapper INaktiv sein).

Anschließend wieder speichern und verlassen mit CTRL-X sowie Bestätigung mit Y(es). Ein Neustart ist nun NICHT erforderlich!

Man mag es kaum glauben: Nun sollte die Kiste (schon?) laufen! Unter feste-ip.net kann man das auch prüfen, d.h. ob der Port 1194 erreichbar ist, indem man auf die beiden Pfeile neben dem Port klickt (dann sollte „Port erreichbar“ erscheinen):



Nun möchte man aber ja auch auf die Geräte hinter der FipBox zugreifen können bzw. den VPN-Tunnel nutzen können (darum macht man das ja alles!). Dies bedingt natürlich wieder etwas (einmalige) Einrichtungsarbeit, und zwar für und an den Endgeräten... Es geht in diesem Sinne weiter wie folgt:

B. **Die Einrichtung des VPN-Zugangs an Endgeräten (hier am Beispiel von iPhone und Macbook Air)**

1. Für die Einrichtung an einem Endgerät brauchen wir im Ergebnis vier Dateien, und zwar „ca.crt“, „client1.crt”, “client1.key” und “fipboxathome.ovpn”. Diese sind auf der FipBox im Verzeichnis /etc/openvpn/easy-rsa/keys gespeichert und müssen von dort auf einen (anderen) Rechner (der ein späteres Endgerät sein kann, aber nicht muss) kopiert werden (am besten in ein separates leeres Verzeichnis). Das sollten wir unter A. 12. bereits erledigt haben. Dann ist Punkt B. 1. damit schon erledigt.

   Falls sich die Dateien noch nicht auf dem gewünschten Zielrechner befinden: Keine Panik. Auch das bekommen wir - allerdings etwas umständlicher - hin. Erst mal am Zielrechner ein leeres Verzeichnis aussuchen oder erstellen. Für den nun notwendigen Transfer gibt es eine Reihe von Möglichkeiten. Wir brauchen aber erweiterte Rechte (also „sudo su“ hilft!). Es geht dann z.B. so:

   a) Die Dateien kann man mittels USB-Stick vom Pi auf den Rechner in das Zielverzeichnis kopieren (vorher mit „sudo su“ erweiterte Rechte bekommen; im Übrigen gehe ich hier auf das konkrete „wie“ ein, ggfs. dazu bitte Tante Google bemühen).

   oder

   b) unter Verwendung des Befehls „scp“. Dann müssen erst mal (temporär) die Rechte des Verzeichnisses /etc/openvpn/easy-rsa/keys geändert werden

und dann eine Abmeldung von der Box erfolgen; das geht mit folgenden Befehlen im Terminal (d.h. an der noch mit SSH verbundenen FipBox!):

- **sudo su**
- **chmod 777 /etc/openvpn/easy-rsa/keys**
- **exit**
- **exit**

Danach ist man von der FipBox erst mal abgemeldet (das soll auch so sein!). Nun gibt man im Terminal (also am Rechner [jedenfalls unter Mac und auch Linux], nicht an der Box) folgende vier Befehle ein, mit dem die Dateien dann herübergeholt werden (wobei natürlich die IP-Adresse der Box und die Ziel-Verzeichnisangaben einzusetzen/anzupassen sind):

- **sudo scp pi@1.1.1.1:/etc/openvpn/easy-rsa/keys/ca.crt /Users/XXX/Desktop/VPN-Daten/**
- **sudo scp pi@1.1.1.1:/etc/openvpn/easy-rsa/keys/client1.crt /Users/XXX/Desktop/VPN-Daten/**
- **sudo scp pi@1.1.1.1:/etc/openvpn/easy-rsa/keys/client1.key /Users/XXX/Desktop/VPN-Daten/**
- **sudo scp pi@1.1.1.1:/etc/openvpn/easy-rsa/keys/fipboxathome.ovpn /Users/XXX/Desktop/VPN-Daten/**

Jeweils sind nun - soweit abgefragt - die Passwörter des Rechners und/oder der FipBox einzugeben. Es sollten dann die vier Dateien auf dem gewünschten Rechner im angegebenen Verzeichnis sein. Prima! Sicherheitshalber sollten wir allerdings die Verzeichnisrechte wieder zurückstellen. Das geht - nach erneuter Anmeldung auf der FipBox mittels ssh - dann so:

- **sudo su**
- **chmod 700 /etc/openvpn/easy-rsa/keys**

Auf weitere Möglichkeiten gehe ich jetzt nicht ein (sprengt den Rahmen). Das Kopieren sollte zu schaffen und nun schon erledigt sein, damit es weitergehen kann...

2. Nachdem wir die vier unter 1. genannten Dateien erfolgreich kopiert haben, sind daran teilweise noch Änderungen erforderlich, die wir mit einem beliebigen Texteditor vornehmen können (ich habe am Mac dafür Textmate verwendet).

   Als Erstes öffnen wir nun die Datei „client1.key“. Sie ist (ganz) leer und nun von uns zu befüllen, und zwar mit dem oben unter A. 12. gesicherten privaten Schlüssel, d.h. wir kopieren diesen einfach in die leere Datei, so dass sie in der 1. Zeile mit „-----BEGIN PRIVATE KEY-----„ beginnt und in der letzten Zeile mit „-----END PRIVATE KEY-----„ endet. Speichern, schließen und fertig.

   Danach öffnen wir noch die Datei fipboxathome.ovpn mit einem Texteditor. Der Inhalt sollte dann etwa so aussehen:

```
dev tun
client
proto tcp
# Tragen Sie hier bitte die Portmapperadresse OHNE DOPPELPUNKT ein !
remote XXXXXXXXX.feste-ip.net YYYYY
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client1.crt
key client1.key
```

comp-lzo
verb 3

Die gelben Markierungen bei der mit „remote“ beginnenden (5.) Zeile sind zu bearbeiten. Hier tragen wir jetzt die von feste-ip.net zur Verfügung gestellte IPV4-Mapping-Adresse ein (im Bild unter A. 10 ganz rechts!). Speichern, schließen und fertig.

3. Die nun fertigen vier Dateien „ca.crt“, „client1.crt”, “client1.key” und “fipboxathome.ovpn“ enthalten nun die notwendigen Informationen, mit der ein sog. Client (also ein Endgerät) sich mit unserem geschaffenen VPN-Server verbinden und damit Pakete durch den VPN-Tunnel schicken kann. Das müssen wir dem Endgerät allerdings noch beibringen... Dies habe ich wie folgt geschafft, wobei nach Endgerät zu unterscheiden ist:

   a. Macbook Air:
      Unsere (vier) Dateien müssen erst mal auf das gewünschte Endgerät gebracht werden (logisch, oder?). Dazu gibt es (wenn es nicht ohnehin schon aus Ziffern 1 und 2 das gewünschte Endgerät ist) viele Wege, z.B. mittels Kopieren auf bzw. von einem USB-Stick.
      Dem Mac ist nun mitzuteilen, dass es ein VPN gibt, dass er auf Wunsch benutzen soll. Leider kann man das nicht einfach in den Einstellungen erledigen, weil die dort möglichen Protokolle nicht passen. Ich habe mir daher die Anwendung „Tunnelblick“ installiert. Das sollte soweit selbsterklärend sein. Anschließend zieht man die Datei „fipboxathome.ovpn“ auf das neue Icon der Anwendung, wodurch die anderen drei Dateien (im selben Verzeichnis!) automatisch mitgenommen werden. Fertig. Nun kann man sich schon mit dem VPN verbinden. Ab geht der Spaß...

   b. iPhone (7):
      Um unsere vier Dateien auf das iPhone zu bringen, bedarf es (da man da ja nicht einfach etwas draufkopieren kann) eines Umwegs, z.B. über iTunes. Außerdem benötigen wir zusätzlich eine App, die das VPN für uns im iPhone ordnungsgemäß und vor allem funktionierend „anmeldet“.In Punkto App habe ich mich für die kostenlose App „OpenVPN“ entschieden, die also zunächst zu installieren ist (andere könnten auch gehen, habe ich aber nicht getestet). Nach Start der App erhält man einige Optionen dazu angezeigt, wie man die nötigen Dateien auf das Smartphone bringen kann. Ich habe mich für den Weg über iTunes entschieden, also das iPhone verbunden, iTunes gestartet, in iTunes unter Dateifreigabe die App OpenVPN ausgewählt und dort (für eben diese App) unsere vier Dateien hinzugefügt. Ist dies erfolgt, kann man in der App (und später dann auch über die Einstellungen) die begehrte VPN-Verbindung starten. Im Erfolgsfall erscheint als Belohnung bzw. Hinweis in der Statusleiste ein schönes umrandetes „VPN“. Schon geht auch hier der Spaß los...

Jetzt sollte alles laufen!

Ziel mit Punktladung erreicht. Ich bin (sehr) zufrieden, tolle Sache!

C. Hinweise und Praktische Nutzung

1. Bei Verwendung von Windows und/oder Android dürften sich in Bezug auf Teil A. praktisch keine Änderungen ergeben (betrifft ja die Box!), wobei man die SSH-Verbindung zur Box z.B. über PUTTY herstellen kann. Getestet habe ich all das mangels Bedarf aber nicht. Das Einrichten der VPN-Verbindungen auf Windows und/oder Android dürfte sich über Tante Google unschwer erschließen lassen.

2. Durch den Eintrag „duplicate-cn“ in der Datei /etc/openvnp/server.conf kann man mit dem einen Schlüssel gleich mehrere Verbindungen aufbauen. Für mich ist das ok, da ich alle nur selbst verwende. Falls Zugriffe (anderer/mehrerer Personen?) mit separaten Client-Keys erfolgen sollen, müssten diese noch auf der FipBox erstellt/generiert werden und die Dateien dann - wie unter B. beschrieben, kopiert, geändert und auf das gewünschte weitere Endgerät gebracht werden. Näheres findet man dazu in den Dokumentationen zu OpenVPN im Netz.

3. Am Macbook (Air) nutze ich nun, wenn ich den VPN-Tunnel aktivieren möchte, die Anwendung „Tunnelblick“ und klicke dort auf Verbinden. Der Aufbau dauert einige wenige Sekunden und schon „ist man drin“.

4. Mit dem iPhone ist es ähnlich einfach: Bei Bedarf schaltet man das VPN über die App „OpenVPN“ oder über die Einstellungen ein. Nach einigen Sekunden erfolgt auch hier die Verbindung (und das Symbol in der Statusleiste).

5. Wenn man den VPN-Tunnel nutzt, kann man tatsächlich das gesamte Heimnetz erreichen. Gebe ich z.B. die IPV4-Adresse meines Routers (der Fritzbox) im Heimnetz an, komme ich problemlos auf dessen Oberfläche. So klappt das auch mit anderen Geräten. Was indes (bei mir jedenfalls derzeit) nicht geht, ist die (heim-)netzinterne Auflösung von Namen - was mich aber auch nicht weiter stört. Natürlich kann man nun auch ins über das VPN ins Internet. Eine WhatIsMyIP-Abfrage ergibt dann, dass es sich um einen Unitymedia-Anschluss handelt, auch wenn ich mit dem verwendeten Endgerät im Mobilfunknetz oder sonstigen Netzen weile.

6. Um die SD-Karte der Box zu schonen, kann man noch die Anzahl der Schreibvorgänge reduzieren. Wer möchte, findet dazu hier detailliertere Informationen: http://www.feste-ip.net/fip-box/basic/fip-box-sd-karte-schonen/ Im Ergebnis hat man dann die im Kern selbsterklärenden weiteren Befehle „schreibschutz_an“ un. „schreibschutz_aus“. Zu beachten ist dabei nur, dass man bei Änderungen jeweils zuvor den Schreibschutz deaktivieren muss - und ihn danach wieder einschalten sollte.

7. Nicht vergessen sollte man natürlich, sich die Einstellungsdaten und Keys zu sichern sowie vom Pi (der FipBox) - so denn alles läuft - zu sichern - also ein Backup zu machen. Auch hier gibt’s viele Wege nach Rom. Man kann das z.B. wie hier (http://www.feste-ip.net/fip-box/basic/fip-box-backup-restore/) beschrieben automatisiert machen, oder man entnimmt die SD-Karte aus der Box und sichert den Inhalt der Karte mit einem entsprechenden Programm am Rechner. Dazu finden sich genügend Tools und Anleitungen im Netz (z.B.

für/mit ApplePi-Baker am Mac), so dass ich hier auf eine Beschreibung verzichte.

8. Was sonst noch geht bzw. sinnvoll erschien (mir jedenfalls):

   a. Um beim Einloggen per SSH als Status gleich angezeigt zu bekommen, welche Portmapper pp. an der Box aktiv sind und welche Temperatur die CPU des Pi hat, habe ich noch folgende Änderungen vorgenommen:

      aa. Anpassung der Datei /etc/fipbox mit dem Befehl „sudo nano /etc/fipbox“ (mit anschließendem Speichern über CRTL-X und Bestätigung mit Y und ENTER), so dass die Datei dann folgenden Inhalt hat (natürlich unbedingt die IP der Box in der letzten Zeile anpassen!!!):

```
################################################################################
# FIP-Box Einstellungen                                                        #
################################################################################
#                                                                              #
# Diese Datei steuert mit einfachen Konfigurationszeilen die FIP-Box           #
#                                                                              #
# Um einen Portmapper zu erstellen tragen Sie die Daten bitte ent-             #
# sprechend der Beispielszeile mit #PORTMAPPER# ein.                           #
#                                                                              #
# Den Updateclient fuer das dynamische DNS koennen Sie im Sinne der Bei-       #
# spielszeile mit #DNS# aktivieren.                                            #
#                                                                              #
# Wenn Sie mit den Einstellungen fertig sind druecken Sie [STRG][X] [Y] [ENTER]#
# um die Einstellungen zu speichern.                                           #
#                                                                              #
# Ein Neutstart der Box ist dann nicht erforderlich!                            #
#                                                                              #
################################################################################

#INAKTIV#VPNMAPTCP#80#192.168.12.3#80
#INAKTIV#VPNMAPUDP#53#192.168.12.3#53

#INAKTIV#DNS#ihr-name.feste-ip.net#HOSTID#Passwort
#INAKTIV#PORTMAPPER#IPv6Port#IPv4ZielIP#IPv4Port
#AKTIV#PORTMAPPER#1194#IP der Box#1194
```

      bb. Anpassung der Datei /etc/bash.baschrc mit dem Befehl „sudo nano /etc/bash.bashrc “. Folgender Text ist am Ende anzuhängen (Achtung: die mit echo -e beginnende Zeile 9 ist etwas länger und enthält keinen Zeilenvorschub!):

```
echo ""
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "IP-Adressen: %s\n" "$_IP"
fi
echo ""
grep "#AKTIV#" /etc/fipbox
echo ""
echo -e "\033[1;33mCPU-Temp:            \033[0m"   `cat /sys/class/thermal/ther-
     mal_zone0/temp | awk '{printf("%.1f\n", $1/1000)}'` "°C"
echo ""
```

      Anschließend - wie üblich - Speichern über CRTL-X und Bestätigen mit Y und ENTER.

   Wenn aa. und bb. erledigt ist, sollte sich die Box beim Anmeldung per SSH brav melden und dabei die aktuellen IP-Adressen (der Box), die aktiven Portmapper pp. und die aktuelle CPU-Temperatur anzeigen.

   Die Anzeige erhält man auch, wenn man als Befehl einfach „bash“ eingibt.

b. Besteht Bedarf, andere konkrete Geräte im lokalen Netz direkt von außen ansprechen zu können (sich also nicht erst via VPN mit dem lokalen Netz zu verbinden), kann man dies nun recht leicht realisieren. Bei Feste-ip.net können ja (ohne weitere Kosten/Credits) für den einmal erstellen universellen Portmapper noch weitere 11 Ports zur Weiterleitung (an unsere FipBox) angegeben werden. Dazu bestimmt man dann also den/die weitere/n Port/s (wobei die Zahlen ziemlich egal sind, sich eben nur nicht mit schon vergebenen oder zu vergebenden überschneiden dürfen) und gibt diese dann unter Portmapper bearbeiten auf feste-ip.net zusätzlich ein. Man erhält dann eine weitere IPV4-Mapping-Adresse. Diese brauch man, um das gewünschte Gerät (etwas später) von außen ansprechen zu können.

   Nun kommen also Anfragen/Zugriffe auf den/m zusätzlich an die FipBox weitergeleiteten Port/s dort an, müssen von der Box aber auch entsprechend behandelt werden. Also ist der Box mitzuteilen, was genau sie damit machen soll. Dies erledigen wir nach Anmeldung an der Box (per SSH wie üblich) mit dem Befehl fipedit, wobei wir für jeden weiteren Port einfach eine weitere Zeile nach dem Schema

   #AKTIV#PORTMAPPER#111#Lokale IP-Adresse#80

   einfügen. Zwischen den Rauten steht also als Erstes logischerweise (ich denke hinreichend selbsterklärend) AKTIV, dann die Art = PORTMAPPER, dann der Port (eben dieser muss jetzt mit der Einstellung bei feste-ip.net identisch sein; schließlich soll die FipBox ja genau darauf jetzt auch hören/reagieren), dann die (lokale) IP-Adresse (an die die Weiterleitung intern erfolgen soll, also des Endgerätes; z.B. eine Webcam) und als Letztes der Port, auf dem unser Endgerät (z.B. die Webcam) die Anfrage (tatsächlich) erwartet.

   Bitte beachten: Da die Weiterleitung an die lokale IPV4-Adresse des Endgeräts erfolgt, wäre es ja jetzt zu dumm, wenn das Gerät von unserem Router - warum und wann auch immer - eine andere lokale IPV4-Adresse erhält. Das Ergebnis wäre dann, dass die Weiterleitung ins Leere oder sonst wo hinliefe. Also sollte auch für erreichbar gemachte Endgeräte von unserem Router (je) eine feste (also immer gleiche) lokale IPV4-Adresse vergeben werden. Dies erledigen wir in der FritzBox unter Heimnetz - Heimnetzübersicht - in den Details zur Box durch Anhaken. Dort heißt es: „Diesem Netzwerkgerät immer die gleiche IPv4-Adresse zuweisen“.

   Bingo: Über die weitere/n IPV4-Mapping-Adresse/n (von feste-ip.net) ist/sind nun das/die Endgerät/e von außen erreichbar!

Im Übrigen bleibt zu hoffen, dass ich nichts Wichtiges vergessen/übersehen habe...

Viele Grüße
Bernd