

Anbindung eines MMDVM-Repeater (auch für duplex-fähige Hotspots) an DMR-OL unter Verwendung des DMRGateway

Um ein MMDVM-basierendes DMR-Relais, ebenfalls Duplex-Hotspots/microRepeater, an das DMR-OL als zusätzliches DMR-Network anzubinden, soll die folgende Anleitung diese Möglichkeit aufzeigen.

Allgemeines

DMR-OL bedeutet **DMR-O**berlausitz**L**ink. Dahinter versteckt sich ein ganzes Team: Ronny/DH5UW, Horst/DG1UHS, Ronny/DG3VU, Rüdiger/DL1RKW und ich, Heiko/DL1BZ. Weiterhin ist kooperativ der DARC-OV S08 (Görlitz) Teil des DMR-OL-Netzwerks, das ist der Betreiber von DB0GRZ.

Das DMR-OL Netzwerk kann aktuell nur MMDVM-basierende Systeme anbinden, Grund ist die alleinige Unterstützung des sog. HB-("Homebrew-")Protokolls, welches von den MMDVM-Systemen zur IP-Vernetzung eingesetzt wird. Mit kommerziellen Repeatern wie MOTOROLA DR3000 können wir derzeit keinen Zugang bereitstellen, für HYTERA wie das RD-985 ginge das unter Zuhilfenahme des HyteraGW, einer Koppelsoftware von den DMRplus-Entwicklern für den Raspberry Pi, die das HYTERA-eigene IPSC-Protokoll in das HB-Protokoll konvertiert und damit dahinter den Einsatz MMDVM-basierender Software wie dem DMRGateway ermöglicht.

Als Software des DMR-OL-Masterservers kommt die Open-Source-Software **Hblink3** (<https://github.com/n0mjs710/hblink3>) und **HBmonitor** (deprecated) von Cort/N0MJS zum Einsatz, welche sich für unseren Zweck hervorragend eignen.

Grundsätzlich stellt das DMR-OL-Netzwerk die Talkgroups **TG70000-TG79999 in beiden Timeslots** und die **TG7 im Timeslot 2** zur Verfügung. Die TG7 stellt dabei eine Art Äquivalent zum BrandMeister-Regio-Cluster, der dort mittels TG8 umgesetzt wird, dar. In der TG7 des DMR-OL-Netzwerks sind dessen Repeater DB0OLL/NLS/SPB/GRZ/FX und diverse Hotspots der User statisch zusammengeschaltet. Gleichzeitig existiert eine Brücke zum C4FM/YSF-Reflektor DE OBERLAUSITZ, die eine mode-übergreifende Kommunikation zwischen DMR und C4FM ermöglicht. Zumindest die TG7 ist entsprechend in den Codeplugs der DMR-Geräte hinzuzufügen - sowohl als TX-Group als auch (falls erforderlich, geräteabhängig) in der entsprechenden RX-Group-List, um diese hörbar zu machen.

Technische Informationen DMR-OL und Zugang

Das DMR-Netzwerk DMR-OL wurde primär dafür entwickelt, hier in der Region Oberlausitz einen Repeaterverbund von derzeit vier DigitalVoice-Repeatern DB0OLL, DB0NLS, DB0GRZ und DB0SPB und einigen Hotspots von Usern unserer Region, die keinen dieser Repeater direkt erreichen können, miteinander zu vernetzen, ohne die großen DMR-Netzwerke wie BrandMeister oder DMRplus verwenden zu müssen. Wir haben nicht den Anspruch, ein weiteres großes DMR-Netzwerk aufzubauen. Vielmehr soll es als Ergänzung zu den bestehenden DMR-Netzen gedacht sein - klein aber fein und vor allen Dingen möglichst einfach in der Nutzung - vor allen Dingen aber stabil und verlässlich im 24/7-Betrieb. Desweiteren basiert der gesamte Master auf Open-Source-Software, was wir grundsätzlich bevorzugen. Zusätzlich zu dem DMR-Master DMR-OL betreiben wir einen C4FM/YSF-Reflektor DE OBERLAUSITZ (#13100), der ebenfalls in den Verbund permanent eingebunden ist und

über eine Brücke mit der TG7 ständig verbunden ist.

DMR-Masterserver-Adresse: dmr02.bzsax.de

Port: 5555/UDP

Passwort: passw0rd (das 0 ist die Ziffer Null!)

C4FM/YSF-Reflektor: DE OBERLAUSITZ (numerisch 13100)

verwendete Talkgroups TS1: 70000-79999

verwendete Talkgroups TS2: 7, 70000-79999

Zugangsprotokoll: ausschließlich HB-Protokoll/MMDVM-Protokoll

Zugangsvoraussetzungen: MMDVM-basierende Repeater und Hotspots (duplex)

Sysop/Administration Server DMR-OL und Kontaktperson: [Heiko/DL1BZ](#)

Die Anbindung an DMR-OL kann so erfolgen, das keine Behinderung oder Einschränkung beim Zugang zu anderen Netzen wie BrandMeister oder DMRplus erfolgen muss. Diese bleiben weiterhin vollständig nutzbar. Das ist auch der Grund, warum wir uns für die TG7 im TS2 entschieden haben, da diese TG in den Netzen BrandMeister und DMRplus dort (im TS2) nicht in Benutzung ist.

Hinweise: Derzeit werden von DMR-OL **nur GroupCalls** verarbeitet, **PrivateCalls oder Datendienste wie TMS können über DMR-OL z.Z. nicht genutzt werden.** Kommerziellen MOTOROLA-Repeater können wir aktuell leider keinen Zugang anbieten (da proprietäres IPSC-Protokoll), kommerziellen HYTERA-Repeater wäre der Zugang unter Einsatz des HyteraGW (Protokollkonverter HYTERA \leftrightarrow MMDVM) möglich. DMR-Systeme mit DMRGateway werden voll supportet.

Datenverarbeitung und Datenschutz: Das DMR-OL-Netz verwendet ausschliesslich offizielle, jedem zugängliche Datenquellen der Registrierstellen, speziell die von radioid.net. Zusätzlich wird eine Last-Heard-Liste generiert, die immer die ca. letzten 500 User listet, die über das System DMR-OL liefen. Diese Liste wird einmal täglich bereinigt, d.h. es wird um 0.00 Uhr auf genau 500 Einträge (ca. 1 1/2 Tage) gekürzt. Logdaten des DMR-Masters werden ebenfalls täglich um 0.00 Uhr vollständig gelöscht. Erfasst werden ausschliesslich Zeit, Dauer, DMR-Id User und DMR-Id Repeater der Übertragung. Diese dienen ausschliesslich zur Funktionsüberwachung des Systems bzw. zur Fehlersuche und -eingrenzung. Sonstige persönliche Daten werden nicht verwendet oder abgefragt.

Alle angeschlossenen Systeme an DMR-OL nutzen die TG7 im Timeslot2, in diese TG7 ist ebenfalls der YSF-Reflektor DE OBERLAUSITZ als Brücke ständig aufgeschaltet. Die Talkgruppen 70000-79999 werden weitgehend für Tests oder Experimente in Sachen DMR oder Brücken in andere Netze genutzt. Der DMR-Master DMR-OL befindet sich in einem Rechenzentrum, Systembasis ist ein Linux/Debian 64bit auf einem dedizierten Server. Es besteht ebenfalls eine fix geschaltene Brücke zum BrandMeister-DMR-Netz, dort kann die TG262907 benutzt werden, um mit der TG7 des DMR-OL kommunizieren zu können.

Wir verstehen uns als Open Network für Funkamateure, die Nutzung von DMR-OL steht grundsätzlich jedem Funkamateur oder Relaisbetreiber offen - eine gültige Amateurfunklizenz und eine DMR-Id ist dafür natürlich erforderlich, Nachweise müssen aber grundsätzlich nicht erbracht werden. Wir arbeiten ausschliesslich auf Vertrauensbasis. Vorschriften oder Regeln gibt es keine - nur bei Missbrauch, der sich ausserhalb der Amateurfunkgesetze und -vorschriften bewegt oder massiv gegen den Hamspirit verstösst und andere Nutzer behindert oder anhaltend stört, behalten wir uns

Zugangsbeschränkungen oder -sperrungen vor. Das sollte aber die absolute Ausnahme bleiben, da wir Restriktionen aus Prinzip ablehnen. Wir versuchen, soweit es unsere Zeit erlaubt, Support für User und Relaisbetreiber zu leisten, allerdings beschränkt auf die Nutzung unseres DMR-OL Netzwerkes und seiner angekoppelten Dienste - ein Anspruch darauf besteht aber grundsätzlich nicht. Wir sind auch kein kommerzieller Dienstleister, die Nutzung von DMR-OL ist und bleibt natürlich kostenlos - also bitten wir auch immer um etwas Geduld, sollten mal Lösungen von Problemen etwas mehr Zeit in Anspruch nehmen. Wir haben alle Familie und einen Job - vergesst das bitte nicht.

Support und Hilfestellungen bei der Konfiguration zum Betrieb am DMR-OL

Unser System ist bei DMR auf das HB-/MMDVM-Protokoll beschränkt, weiterhin ausschließlich auf die Digi-Modi DMR und C4FM/YSF. Für D-STAR gibt es grundsätzlich keinen Support, da dieser Mode bei uns nicht zum Einsatz kommt und auch nicht kommen wird.

Grundsätzlich gehen wir von MultiNet-DMR-Betrieb aus, d.h ihr bleibt primär in einem der Hauptnetze BrandMeister und/oder DMRplus, wollt aber DMR-OL als weiteres Zusatznetz ebenfalls nutzen.

Die Umsetzung erfolgt grundsätzlich mittels des zur MMDVM-Plattform gehörenden Tools [DMRGateway\(G4KLX\)](#) (bei PiStar bereits enthalten).



WICHTIG: Wir können **keinen Support für Beta- oder RC-Versionen von PiStar** leisten. Diese sind Test-/Prüfversionen und keine offiziellen STABLE-Releases - können also durchaus Fehler und Probleme aufweisen, mit deren Ursachen wir uns aus Zeitgründen nicht auseinandersetzen können. Die derzeit aktuelle offizielle STABLE-Version [PiStar](#) ist nach wie vor die **V3.4.17 vom 20.1.2019** ! Leider unterstützt der dort eingesetzte Linux-Kernel nicht den Raspberry Pi 3B+/4 - *bootet* also per default *nicht* auf den genannten Pi's.

Desweiteren kommen bei [PiStar](#) oft *modifizierte* Versionen des MMDVMHost, DMRGateway und anderer MMDVM-Tools zum Einsatz. Diese entsprechen nicht mehr in der Codebasis den eigentlich offiziellen Versionen von [G4KLX](#) und beinhalten oft Erweiterungen/Modifikationen in deren .ini-Dateien oder Binär-Paketen und sind als sog. **Fork** zur eigentlichen offiziellen Codebasis zu sehen. Wir sehen ein solches Vorgehen als durchaus problematisch, da es bei unseren Beispielkonfigurationen möglicherweise zu *Fehlfunktionen führen kann*. Wir supporten also ausschließlich die offiziellen Original-Versionen von [G4KLX](#), versuchen aber - wo möglich - Hinweise auf [PiStar](#)-spezifische Anpassungen zu geben - sofern uns diese bekannt sind oder werden.

Direkten Support (von uns getestet und verifiziert), also direkte Unterstützung von Systemen, die zu Einsatz kommen, können wir derzeit ausschließlich leisten für:

- MMDVM-basierende "echte" Repeater
- MMDVM-basierende Duplex-Hotspots - häufig mit PiStar als Basis-System betrieben - wobei wir primär die Duplex-Hotspots bei DMR klar bevorzugen und empfehlen, da es sich quasi um

vollwertige Repeater wie die "Großen" handelt und identische Möglichkeiten zur Verfügung stehen. Wir nennen diese auch schlicht micro-Repeater, da es besser deren Funktionalitäten abbildet.

- MMDVM-basierende Simplex-Hotspots - häufig mit PiStar als Basis-System betrieben - hier ist aber zu beachten, das speziell bei DMR nur ein einziger Timeslot existiert, was eine flexible Konfiguration leider etwas einschränkt und spezielle Anpassungen besonders am Regelwerk des DMRGateway erfordert

Indirekten Support (ohne Funktionsgarantie !) - da hier zum Testen leider nicht verfügbar:

- OpenSpot 1,2 und 3
- Blue-DV oder ähnliches
- sonstige "Speziellösungen", die zwar MMDVM sprechen, aber deren Spezifika wir (noch) nicht genau kennen
- Zugang mit kommerziellen HYTERA-Repeatern (RD625, RD985), wobei hier der Einsatz des Zusatztools [HyteraGW](#) (von Kurt/OE1KBC) *erforderlich* ist, um das DMRGateway nutzen zu können

Keinen Support - da systembedingt nicht unterstützt:

- DV4mini, DV4home oder ähnliche Lösungen **mit altem Dongle-Protokoll**
- kommerzielle MOTOROLA-Repeater (u.a. DR3000) - diese sprechen ein anderes Protokoll (IPSC) und können derzeit keinen direkten Zugang bei DMR-OL bekommen, da wir für das MOTOROLA-IPSC-Protokoll **keine Schnittstelle bereitstellen können** (diese ist *nicht offengelegt* und unterliegt den Lizenzbedingungen von MOTOROLA)

Das bedeutet nicht, das andere Lösungen, die aktuell auf MMDVM basieren (wie auch die DV4-Lösungen mit aktueller Software), keinen Zugang bei uns erhalten können. Nur in diesem Fall bitte mit uns in Verbindung setzen, wir werden versuchen, hier gemeinsam eine Umsetzung zu erarbeiten. Nur können wir es selbst hier nicht verifizieren und damit auch keine Beispiel-Konfigurationen vorab zur Verfügung stellen.

Dashboards des DMR-OL-Netzwerks

Webdashboard des DMR-Masters DMR-OL: <https://dmr02.bzsax.de:8488/>

Webdashboard des YSF/C4FM-Reflektors DE OBERLAUSITZ: <https://ysf02.bzsax.de/ysf-ol/>

Verwendung des DMRGateway von G4KLX

Das DMRGateway (<https://github.com/g4klx/DMRGateway>) als Teil der Open-Source-MMDVM-Plattform, die weitgehend von Jonathan/G4KLX entwickelt wird, bietet die Möglichkeit, an MMDVM-basierenden Systemen MultiNet-Betrieb, d.h. mehrere DMR-Netzwerke gleichzeitig, zu ermöglichen. Gesteuert wird das alles über eine Art Regelwerk, welches in der *dmrgateway[.ini]* entsprechend definiert wird. Es sorgt dafür, wie eine Art Router entsprechende TGs oder PCs in den entsprechenden Timeslots den connecteten DMR-Netzwerken zuzuordnen. Derzeit sind bis 5 DMR-Netzwerke und ein XLX möglich.

Genau dieser Möglichkeit bedienen wir uns zur Anbindung des DMR-OL-Netzwerks als *weiteres, zusätzliches DMR-Netzwerk* neben z.B. BrandMeister und/oder DMRplus.

===== Umsetzung am Beispiel eines DMR-Repeater oder Duplex-Hotspots mit PiStar =====

HINWEIS: Grundsätzlicher Umgang mit SSH, Linux (dessen Shell und einem Editor wie nano) wird ab hier vorausgesetzt !

Weiterhin geht die beschriebene Konfiguration davon aus, einen vollwertigen Repeater mit beiden Timeslots zur Verfügung zu haben. Für Simplex-Hotspots müsste die Konfiguration angepasst werden, da nur ein einziger Timeslot verfügbar ist.

Da die Nutzung von DMR damit eingeschränkt wäre, gehe ich auf diese Option in der folgenden Anleitung nicht weiter ein (bitte ggf. anfragen).

Das folgende Beispiel nutzt 2 Netzwerke: BrandMeister als primäres, DMR-OL als sekundäres, zweites DMR-Netzwerk. Es ginge aber natürlich auch mit DMRplus als primäres Netzwerk - mit den entsprechenden geänderten Anpassungen in der DMRgateway[.ini].

Im Grunde funktioniert das so:

Der MMDVMHost connected das DMRGateway wie einen eigenen Master, dann wiederum connectet das DMRGateway die einzelnen, dort definierten Netzwerke. Das Regelwerk des DMRGateway sorgt für das korrekte Routing von Talkgroups und PrivateCalls an die entsprechenden Netzwerke, wobei Regeln birektional und unidirektional möglich sind, ebenfalls ermöglicht ist ein Rewrite von einem GroupCall zu einem PrivateCall - andersrum aktuell leider nicht.

Vorbereitungen/Voraussetzungen im PiStar:

Verwendung des DMRGateway aktivieren mittels Dashboard/Webinterface unter **Konfiguration:**

DMR Konfiguration	
Einstellung	Wert
DMR Master:	DMRGateway

Da die Konfiguration des DMRGateway unter PiStar über das Webinterface nicht flexibel genug ist, müssen wir das also per SSH auf der Shell tun !

⚠ HINWEIS: Jede spätere Änderung per Webinterface überschreibt dann leider wieder unsere angepasste Konfiguration, daher also die Konfig entsprechend sichern und ggf. wieder zurückspielen.

Login also nun per SSH, von WINDOWS aus machen wir das am besten mit dem freien Tool PuTTY.

```
$ sudo -s
$ rpi -rw
```

Sichern der originalen `/etc/dmrgateway`

```
$ cd /etc
$ cp dmrgateway dmrgateway.org
```

Editieren der `/etc/dmrgateway`

```
$ nano /etc/dmrgateway
```

Anpassen der `/etc/dmrgateway` für unsere Zwecke - also den **gesamten Inhalt zuerst löschen** und durch folgendes **ersetzen**:

```
[General]
RptAddress=127.0.0.1
RptPort=62032
LocalAddress=127.0.0.1
LocalPort=62031
# wenn RuleTrace=1 kann im DMRGateway-Log die Funktion der Regeln geprüeft
werden, bleibt normalerweise AUS
RuleTrace=0
Daemon=1
Debug=0
RFTimeout=20
NetTimeout=20

[Log]
DisplayLevel=0
FileLevel=1
FilePath=/var/log/pi-star
FileRoot=DMRGateway

[Voice]
Enabled=1
Language=de_DE
Directory=/usr/local/etc/DMR_Audio

[Info]
Enabled=0

[XLX Network]
Enabled=0

[DMR Network 1]
Enabled=1
Name=DMR-0L
Address=dmr02.bzsax.de
Port=55555
Password="passw0rd"
#
# Regel bei TGRewrite ist immer so: TGRewrite=RFSlot,RF-
TG,NetzwerkSlot,Netzwerk-TG,Bereichsangabe
# damit sind Anpassungen oder Umsetzungen zwischen RF- und Netzwerkseite
*direkt* HF-seitig am Repeater moeglich - ohne Anpassungen am angekoppelten
Netzwerk machen zu muessen
# die Regel gilt automatisch bidirektional - wirkt also in _beide_
Richtungen: ankommend/gehend
# Hinweis: alles im TG-Bereich 70000-79999 in beiden Slots zzgl. der TG7/nur
TS2 wird immer nach DMR-0L weggeroutet !
# Der Rest bleibt da wo er sein soll und wird nicht angefasst, z.B. beim BM
oder DMRplus oder was auch immer, je nach Konfig
```

```
# DMR-OL verarbeitet *nur Gruppenrufe*
- keine PrivatCalls, TMS/SMS oder anderes !
#
# TG7 auf TS2 nach DMR-OL
TGRewrite=2,7,2,7,1
# alternativ TG7 auf TS1 nach DMR-OL - bitte nur eine der Optionen verwenden
durch Setzen/Entfernen des Kommentarzeichens
# BEACHTET: BEI SIMPLEX-Hotspots folgende Zeile auskommentiert lassen !
# TGRewrite=1,7,2,7,1
# BEACHTET: BEI SIMPLEX-Hotspots folgende zwei Zeilen auskommentiert lassen !
# TG70000-79999 auf TS1 nach DMR-OL
# TGRewrite=1,70000,1,70000,10000
# TG70000-79999 auf TS2 nach DMR-OL
TGRewrite=2,70000,2,70000,10000
Location=0
Debug=0
# DMR-Id Repeater bzw. Hotspot ab PiStar>= 4.0 hier nochmals hinzufuegen
# Id=123456

[DMR Network 2]
Enabled=1
Name=BrandMeister
# Auswahl der BM-Master
# Ziel BM2001 - BM-EU-Master
# Address=bm.afu.rwth-aachen.de
# Ziel BM2621
Address=master1.bm262.de
# Ziel BM2622
# Address=master2.bm262.de
Port=62031
Password="passw0rd"
# ausser den TG unter DMR-OL geht der Rest alles in Richtung BrandMeister
PassAllPC=1
PassAllPC=2
PassAllTG=1
PassAllTG=2
Location=0
Debug=0
# DMR-Id Repeater bzw. Hotspot ab PiStar>= 4.0 hier nochmals hinzufuegen
# Id=123456

[DMR Network 3]
Enabled=0
```

Nun sichern wir am besten nochmals unsere eben geänderte, angepasste */etc/dmrgateway*, falls wir die mal ggf. zurückkopieren müssen (wenn das Webinterface die mal überschrieben haben sollte).

```
$ cp /etc/dmrgateway /etc/dmrgateway.save
```

Jetzt starten wir PiStar einfach neu:

```
$ sudo reboot
```

Anschließend unter <https://dmr02.bzsax.de:8488/> prüfen, ob sich PiStar korrekt am Master DMR-OL angemeldet hat. **Für SIMPLEX-Hotspots bitte wegen anzupassender (individueller?) Konfig bzgl. DMRGateway mit uns in Verbindung setzen !**

Empfohlen: Prüfen der Netzwerkverbindungsqualität mit iperf3 (speziell UDP-Protokoll)

Um die Verbindungsqualität zum DMR-OL-Masterserver prüfen zu können, verwenden wir ein Tool zur Messung von Netzwerkdatenverkehr **iperf3**. Hierzu stelle ich auf unserem DMR-OL-Master einen Testservice bereit. DigitalVoice over IP benutzt das UDP-Protokoll und stellt praktisch die gleichen Anforderungen wie Voice-over-IP ("Internettelefonie"), was im Handling einige Fallstricke mit sich bringen kann. Wir benötigen nicht viel Bandbreite, aber dafür sehr kurze Latenzzeiten und wenig bis gar keine Verluste an Datenpaketen. Als Latenz bezeichnet man die Laufzeit eines Datenpakets von der Quelle, z.B. der Raspberry Pi des Repeaters oder Hotspots, bis zum Ziel, dem DMR-Masterserver.

Wir loggen uns also wieder per SSH in PiStar/Linux ein und installieren erst mal das erforderliche Paket **iperf3**:

Folgendes nur bei Verwendung von PiStar:

```
$ sudo -s  
$ rpi -rw
```

weiter bei PiStar UND Linux:

```
$ sudo apt-get install iperf3
```

NUR PiStar:

Leider lässt der Firewall des PiStar "ab Werk" keine Verbindung zu unserem DMR-Master zu, wenn wir mit **iperf3** testen wollen oder auch **andere Ports** verwenden möchten. Hier müssen wir erst mal ein wenig nachhelfen, damit diese Verbindungen möglich sind.

```
$ sudo -s  
$ rpi -rw  
$ cd /root  
$ nano ipv4.fw
```

In diese Datei *ipv4.fw* tragen wir folgendes ein bzw. fügen folgende Zeilen hinzu (falls es schon eine *ipv4.fw* gibt und Einträge bereits vorhanden sein sollten) und speichern diese danach ab:


```
iptables -I INPUT -s 5.9.59.26 -j ACCEPT
iptables -I OUTPUT -d 5.9.59.26 -j ACCEPT
iptables -I INPUT -s 85.214.61.75 -j ACCEPT
iptables -I OUTPUT -d 85.214.61.75 -j ACCEPT
```

Anschließend aktualisieren wir die Firewall-Settings des PiStar - es bleibt alles auch nach einem Neustart erhalten:

```
$ sudo pistar-firewall
```

Nun wieder PiStar UND Linux:

Nun können wir endlich den Test ausführen, dazu tippen wir folgende Befehlszeile ein:

```
$ sudo iperf3 -u -p 62036 -c dmr.bzsax.de -t 30 -P 2
```

Nun werden für 30s zwei parallele UDP-Datenstreams erzeugt und zum DMR-OL-Master geschickt. Anschließend sollte nach dem Test eine Zusammenfassung angezeigt werden, die wie folgt aussehen sollte:

```
- - - - -
[ ID] Interval          Transfer          Bandwidth          Jitter          Lost/Total
Datagrams
[  4]  0.00-30.00  sec  3.74 MBytes  1.05 Mbits/sec  7.535 ms  0/479 (0%)
[  4] Sent 479 datagrams
[  6]  0.00-30.00  sec  3.74 MBytes  1.05 Mbits/sec  7.484 ms  0/479 (0%)
[  6] Sent 479 datagrams
[SUM]  0.00-30.00  sec  7.48 MBytes  2.09 Mbits/sec  7.510 ms  0/958 (0%)

iperf Done.
```

Das Ergebnis bedeutet: Wir haben insgesamt 958 UDP-Datenpakete gesendet, davon wurden keine verworfen ("lost: 0"). Der Jitter (Schwankungen in der Laufzeit der Datenpakete) ist ebenfalls gut mit ca. 8ms.

Kommt es bei Euch evtl. zu größeren Paketverlusten ("packet loss") oder hohem Jitter, ist mit hoher Wahrscheinlichkeit mit Problemen der Sprachverbindung am Repeater/Hotspot zu rechnen. Das kann sich in fehlenden Durchgängen, digitalen Artefakten ("Klötzchenbildung") oder sogar einem Disconnect vom DMR-OL-Master äußern. Hierbei wäre die Internetverbindung zu überprüfen, ob diese für die Übertragung von Sprache überhaupt geeignet ist.



Hinweis:

Es sollte unbedingt eine Priorisierung ("QoS") von UDP-Paketen am Internet-Router vorgenommen werden - sofern der Internetanschluss von weiteren Geräten genutzt wird. Auch sollte nach Möglichkeit Ethernet gegenüber WLAN immer bevorzugt werden. Von der WLAN-Nutzung auf 2,4GHz rate ich insbesondere in dicht besiedelten Gebieten dringend ab - wenn schon unbedingt WLAN, dann immer auf 5GHz ausweichen, als Norm ist immer 802.11n oder 802.11ac zu wählen.

Optional: Erweiterung um 3. Netzwerk DMR-DL (DMRplus-Netzwerk)

Um DMRplus ebenfalls zu integrieren, ist das DMRGateway um ein drittes Netzwerk - hier [DMR-DL](#) (früher DMR-MARC) - zu erweitern. DMR-DL nutzt die Infrastruktur von DMRplus, ist aber mehr oder weniger als eigenständig und unabhängig von DMRplus an sich in DL zu betrachten. Es gilt aber die Bedienung/Benutzung wie bei DMRplus allgemein. DMR-DL ist einer der Kooperationspartner von DMR-OL. Wir pflegen also enge Kontakte zu DMR-DL und sind diesem Netzwerk freundschaftlich sehr verbunden.

```
[DMR Network 3]
Enabled=1
Name=DMR-DL
Address=ipsc.dmr-dl.net
Port=55555
# entweder
Password="PASSWORD"
# oder
# Password="passw0rd"
#
# Startreflektor 4044 Berlin-Brandenburg in der TG9/TS2, 15min Rueckfall,
# Buchen der TG262 von DMRplus
Options="StartRef=4044;RelinkTime=15;TS1_1=262;TS1_2=0;TS1_3=0;TS1_4=0;TS1_5
=0;TS2_1=0;TS2_2=0;TS2_3=0;TS2_4=0;TS2_5=0;"
# entweder die TG9/TS2 im DMRplus mit deren Reflektoren nutzen
TGRewrite=2,9,2,9,1
# oder deren TG9 alternativ nach TS1 verlagern
# TGRewrite=1,9,2,9,1
# da die TG262 auch bei DMRplus existiert, muessen wir diese mit einer
# anderen TG, hier TG9262, auf der HF-Seite ansprechen
TGRewrite=1,9262,1,262,1
# folgende Regeln sind erforderlich für DMRplus-Anwendungen wie APRS-SMS und
# Reflektor-Steuerung per PrivateCalls
PCRewrite=1,5050,1,5050,10
PCRewrite=2,5050,2,5050,10
PCRewrite=1,9050,1,9050,10
PCRewrite=2,9050,2,9050,10
TGRewrite=1,5066,1,5066,1
TGRewrite=2,5066,2,5066,1
TGRewrite=1,9066,1,9066,1
TGRewrite=2,9066,2,9066,1
TGRewrite=2,9990,2,9990,1
# Reflektoren steuern per PrivateCall, in beiden TS
PCRewrite=1,4000,2,4000,1001
PCRewrite=2,4000,2,4000,1001
TypeRewrite=1,5055,1,9055
TypeRewrite=1,5056,1,9056
TypeRewrite=1,5057,1,9057
```

```
TypeRewrite=1,5058,1,9058
TypeRewrite=1,5059,1,9059
TypeRewrite=2,5055,2,9055
TypeRewrite=2,5056,2,9056
TypeRewrite=2,5057,2,9057
TypeRewrite=2,5058,2,9058
TypeRewrite=2,5059,2,9059
Location=0
Debug=0
# DMR-Id Repeater bzw. Hotspot ab PiStar>= 4.0 hier nochmals hinzufuegen
# Id=123456
```

Zugriff Webdashboard bei DMR-DL (DMRplus): <http://ipsc.dmr-dl.net/ipsc/>

73 Heiko/DL1BZ
Sysop DB0OLL/SPB/NLS/GRZ
Admin DMR-OL

From:
<http://kb.amft-it.de/> - **Amateurfunk - Knowledge Base und Wiki by DL1BZ**

Permanent link:
<http://kb.amft-it.de/doku.php?id=kb-afu:dmrol-relaisanbindung>

Last update: **29.11.2021 18:15**

