

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



Um die wirkliche Verbindungsqualität speziell mit UDP zu testen/prüfen, sind einfache Tools wie *ping* oder *traceroute* **nicht geeignet** !

Diese dienen eher zum Test der Erreichbarkeit eines Ziels, sagen aber wenig bis gar nichts darüber aus, ob die Verbindung wirklich den Ansprüchen (Jitter, Paketverluste, Latenz) genügt ! Speziell die Frage der Latenz, also die Laufzeit, die ein Datenpaket von der Quelle bis zum Ziel oder zurück benötigt, muss so kurz/klein wie möglich sein. Sprache erfordert Echtzeitübertragung !

Um die Verbindungsqualität zu unserem Server prüfen zu können, verwenden wir ein Tool zur Messung von Netzwerkdatenverkehr **iperf3**. Hierzu stelle ich auf unserem Server dmr02.bzsax.de 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 - also dem Server (YSF-Reflektor oder DMR-Master).

Im Folgenden wird der Umgang mittels ssh (Putty oder ähnliches) vorausgesetzt. Das Editieren von Dateien mit einem Texteditor (z.B. nano) wird ebenfalls vorausgesetzt. Bitte die speziellen Hinweise bei Einsatz von PiStar gegenüber "normalen" Linux-Systemen beachten. PiStar arbeitet mit einem Read-Only-Filesystem, wodurch mittels `rpi - rw` die SD-Card in den Schreibmodus versetzt wird. Manchmal ist das erforderlich.

Wir loggen uns also per SSH (z.B. mit Putty) in PiStar/Linux ein und installieren erst mal das erforderliche Paket **iperf3** (per default ist es nicht vorhanden bzw. fehlt):

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 Server 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. Jedoch haben die Entwickler von PiStar [diesen Fall vorbereitet](#), indem man den Firewall um weitere, eigene Regeln ergänzen kann. In diesem, folgenden Fall erlauben wir alle Protokolle und Ports von/zu unserem Server vom Relais aus.

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

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

```
iptables -I INPUT -s 85.214.61.75 -j ACCEPT
iptables -I OUTPUT -d 85.214.61.75 -j ACCEPT
iptables -I INPUT -s 5.9.59.26 -j ACCEPT
iptables -I OUTPUT -d 5.9.59.26 -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 Befehlszeilen (zwei Tests: erste Zeile testet Daten vom Relais zum Server, die zweite Zeile testet Daten vom Server zum Relais) ein:

```
$ sudo iperf3 -u -p 42006 -c dmr02.bzsax.de -t 30 -P 2
$ sudo iperf3 -u -p 42006 -c dmr02.bzsax.de -t 30 -P 2 -R

$ sudo iperf3 -u -p 42006 -c 5.9.59.26 -t 30 -P 2
$ sudo iperf3 -u -p 42006 -c 5.9.59.26 -t 30 -P 2 -R
```

Erklärung der Parameter des Testtools iperf3 :

- u UDP-Protokoll
- p 42006 Port auf dem das Testtool des Servers hört
- c dmr02.bzsax.de Adresse unseres Servers 1 bzw. -c 5.9.59.26 Adresse unseres Servers 2
- t 30 Dauer 30 sec
- P 2 zwei parallele Streams gleichzeitig
- R Reverse, in diesem Fall sendet der Server zum Relais, andernfalls das Relais zum Server (Richtungsumkehr, damit beides - also Senden und Empfangen beim Relais via Netzwerk getestet werden kann)

Nun werden für 30s jeweils zwei parallele UDP-Datenstreams erzeugt und zum Server geschickt (Test 1) bzw. vom Server empfangen (Test 2). Anschließend sollte nach jedem 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 Server 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.

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:iperf&rev=1638303145>

Last update: **30.11.2021 21:12**

