

# Ubuntu-Handy als Internetgateway

## Ubuntu-Handy als Internetgateway

### Konfiguration

Dipl.Ing. Christoph Stockmayer, Hiltpoltstein, [sto@stockmayer.de](mailto:sto@stockmayer.de)

## 1. Einleitung

Die Fragestellung lautete: Wie bekomme ich ein Ubuntu-Smartphone (BQ Aquaris E4.5 UbuntuEdition) dazu, als InternetGateway für mein Notebook zu funktionieren, ohne WLAN via USB-Kabel? Im Internet findet man verschiedene Ansätze, es ist von „rooten“ des Handys zu lesen, tethering ist ein Schlagwort dazu, die USB-Verbindung müsste für das Netzwerk eingerichtet werden und anderes – eine passende App dazu, die alles bewerkstelligt, hab ich allerdings nicht gefunden.

## 2. Vorbereitung

Zur Vorbereitung sind folgende Schritte notwendig:

- USB-Verbindung zwischen Handy und PC
- Das Paket `android-tools` auf dem PC installieren
- Konfiguration von `adb` durch Eintrag des USB-Devices in die Datei `/root/.android/adb_usb.ini` auf dem PC (dies ist allerdings nicht zwingend!):
  - `lsusb`

```
Bus 002 Device 003: ID 0a5c:5801 Broadcom Corp. BCM5880 Secure Applications Processor
with fingerprint swipe sensor
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0c45:64d0 Microdia
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 075: ID 1058:1021 Western Digital Technologies, Inc. Elements 2TB
Bus 003 Device 015: ID 046d:c52f Logitech, Inc. Unifying Receiver
Bus 003 Device 079: ID 2a47:0c02
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```
  - Identifizieren der USB-ID
  - Eintrag in `/root/.android/adb_usb.ini` (ins Heimatverzeichnis der root deshalb, da nachher ein `ppp`-Daemon als root gestartet werden muss):  
`0x2a47`
- Handy in den Entwicklermode stellen:  
Systemeinstellungen → Info zu diesem Gerät → Entwicklermodus ✓

## Ubuntu-Handy als Internetgateway

### 3. Schritte zum Aufbau der Netzwerkverbindung

Nun die Schritte zum Aufbau einer ppp-Verbindung.

Im Internet finden sich Aussagen, dass die USB-Verbindung in den Netzwerkstatus zu stellen sei (was aus meiner Sicht nicht notwendig ist). Dies könnte (falls doch notwendig) so geschehen:

- Öffnen eines Befehlfensters (**Terminal-App** möglicherweise erst installieren)
- mit `sudo android-gadget-service enable rndis` und Eingabe des Passworts (das auch für das Handy selbst verwendet wird) wird der Netzwerkmode eingeschaltet
- später wieder umgeschaltet mit `sudo android-gadget-service disable rndis` und `sudo android-gadget-service enable mtp`
- statt vieler sudo's ist es auch möglich, permanent in die root zu wechseln: `sudo su -`
- statt mühsam auf der virtuellen Handy-Tastatur zu tippen, kann auch mit adb vom PC her auf die Kommandozeile des Handys zugegriffen werden:

```
asterix:~# adb shell
phablet@ubuntu-phablet:~$ sudo su -
[sudo] password for phablet: HandyCode
root@ubuntu-phablet:~#
```

- alternativ kann dies die **TweakGeek-App** bewerkstelligen
- ein permanentes Umschalten in den rc-Skripten ist mir nicht gelungen

Nun muss eine ppp-Verbindung via dieser USB-Schnittstelle eingerichtet werden. Da der adb-Dienst auf dem Handy als normaler Benutzer 'phablet' läuft, ist ein sudo notwendig, das aber normalerweise nach dem HandyCode-Passwort fragt. Dies lässt sich nicht automatisieren, es sei denn, ich erlaube ein sudo mit pppd in der Datei `/etc/sudoers` direkt editiert auf dem Handy oder mit dem Kommando `visudo` (in einer neuen Zeile am Ende der Datei). Zuvor muss das root-Dateisystem in den SchreibMode umgestellt werden:

```
mount -o remount,rw /
```

später wieder zurück:

```
mount -o remount,ro /
```

Dann also der Eintrag in `/etc/sudoers`:

```
phablet ubuntu-phablet = NOPASSWD: /android/system/bin/pppd
```

Nun funktioniert der folgende Befehl ausgeführt auf dem PC:

```
adb ppp "shell:/usr/bin/sudo /android/system/bin/pppd nodetach noauth \
noipdefault /dev/tty" nodetach noauth noipdefault \
notty 10.0.0.1:10.0.0.2
```

Manchmal klappt er bei mir auf Anhieb nicht, dann versuche ich es mehrmals. Als Resultat muss nun auf dem PC ein ppp-Prozess laufen (kontrolliert mit `ps -ef | grep ppp`) und auf dem Handy. Der ppp hat nun auf dem PC ein NetzwerkDevice mit der IP `10.0.0.1` und auf dem Handy eines mit `10.0.0.2` angelegt.

Damit sich der ppp bei Inaktivität nicht ausschaltet, würde ich ab und an einen ping drauf machen:

## Ubuntu-Handy als Internetgateway

```
ping -c 3 10.0.0.2
```

Nun ist noch das Forwarding auf dem Handy einzuschalten:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
echo "1" > /proc/sys/net/ipv4/ip_dynaddr
```

und Firewallregeln für das Weiterleiten der IP-Pakete ins Internet (der externen Schnittstelle des Handys. s. [5]):

```
EXTIF="ccmni0"
INTIF="ppp0"
iptables-restore <<-EOF
*nat
-A POSTROUTING -o "$EXTIF" -j MASQUERADE
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -i "$EXTIF" -o "$INTIF" -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i "$INTIF" -o "$EXTIF" -j ACCEPT
-A FORWARD -j LOG
COMMIT
EOF
```

Am besten erledigt man dies mit einem Skript (hier mit dem Namen router.sh) auf dem Handy, das natürlich dann auch mit Hilfe von `sudo` aufgerufen werden muss und weshalb ein weiterer Eintrag in `/etc/sudoers` fällig wird:

```
phablet ubuntu-phablet = NOPASSWD: /home/phablet/router.sh
```

Dann bin ich fast fertig, mit Default-Route und Eintrag eines Namensservers auf dem PC:

```
route del default (falls zuvor bereits einer gesetzt war)
```

```
route add default gw 10.0.0.2
```

```
in /etc/resolv.conf: nameserver 8.8.8.8 (oder ein anderer DNS)
```

funktioniert die Verbindung, zu testen mit `ping`:

```
ping www.opensuse.org
```

Nun kann auf dem Handy auch ein `ssh`-Dienst gestartet

```
/usr/sbin/sshd -D
```

und vom PC via

```
ssh phablet@10.0.0.2
```

eine Befehlszeile auf dem Handy aktiviert werden.

## Ubuntu-Handy als Internetgateway

### 4.Und alles als Skript

Und das nun alles als Skript, das vom PC gestartet werden kann (Voraussetzungen sind die beiden /etc/sudoers-Einträge im Handy und das Skript router.sh ebenfalls auf dem Handy):

```
#!/bin/sh
#####
#
# Verbindet mit BQ Ubuntu Phone
#
# usage: bqconnect.sh [wlan]
#
# Parameter: wlan - falls Handy im WLAN
#
#           nichts - falls Handy im Netz
#
# Voraussetzung im Handy: Skript 'router.sh' für Firewall
#
#                   Eintrag pppd in /etc/sudoers
#
#                   Eintrag router.sh in /etc/sudoers
#
#
# (co) Stockmayer
#
# 12.08.2015
#
#####
LANG=C

until ps -ef | grep -v grep 2> /dev/null | grep -q ppp 2> /dev/null
do
    adb ppp "shell:/usr/bin/sudo /android/system/bin/pppd nodetach noauth noipdefault \
            /dev/tty" nodetach noauth noipdefault notty 10.0.0.1:10.0.0.2
    sleep 1
done

trap 'kill $sleepPID; cp /tmp/resolv.conf$$ /etc/resolv.conf; \
      route del default 2> /dev/null; exit 1' 2

sleep 1
while true
do
    ping -c 3 10.0.0.2 > /dev/null 2>&1
    sleep 20
done &
sleepPID=$!

adb shell sudo /home/phablet/router.sh ${1:ccmni}0

cp /etc/resolv.conf /tmp/resolv.conf$$
cat > /etc/resolv.conf <<EOF
domain stockmayer.de
nameserver 8.8.8.8
EOF
```

## Ubuntu-Handy als Internetgateway

```
sleep 1
/sbin/route del default 2> /dev/null
/sbin/route add default gw 10.0.0.2

ping -c 3 www.opensuse.org
echo "all ok, network is ready to use ... (end with CTRL-C)"
wait
```

### Das router.sh-Skript auf dem Handy (s. [5]):

```
#!/bin/bash
echo "\n\nLoading simple rc.firewall-iptables..\n"
if test $# = 1
then
    EXTIF="$1"
else
    EXTIF="ccmni0"
fi
INTIF="ppp0"
echo "  External Interface:  $EXTIF"
echo "  Internal Interface:  $INTIF"
echo "  Enabling forwarding.."
echo "1" > /proc/sys/net/ipv4/ip_forward
echo "  Enabling DynamicAddr.."
echo "1" > /proc/sys/net/ipv4/ip_dynaddr
echo "  Clearing any existing rules and setting default policy.."

iptables-restore <<-EOF
*nat
-A POSTROUTING -o "$EXTIF" -j MASQUERADE
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -i "$EXTIF" -o "$INTIF" -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i "$INTIF" -o "$EXTIF" -j ACCEPT
-A FORWARD -j LOG
COMMIT
EOF
echo "\nrc.firewall-iptables done.\n"
```

## 5.Test

Nach dem Start des Skripts (hier `bqconnect.sh`) auf dem PC:

## Ubuntu-Handy als Internetgateway

```
asterix:~ # bqconnect.sh
```

```
Loading simple rc.firewall-iptables..
```

```
External Interface: wlan0
```

```
Internal Interface: ppp0
```

```
Enabling forwarding..
```

```
Enabling DynamicAddr..
```

```
Clearing any existing rules and setting default policy..
```

```
rc.firewall-iptables done.
```

```
PING www.opensuse.org (130.57.66.6) 56(84) bytes of data.
```

```
64 bytes from www.opensuse.org (130.57.66.6): icmp_seq=1 ttl=43 time=194 ms
```

```
64 bytes from www.opensuse.org (130.57.66.6): icmp_seq=2 ttl=43 time=299 ms
```

```
64 bytes from www.opensuse.org (130.57.66.6): icmp_seq=3 ttl=43 time=193 ms
```

```
--- www.opensuse.org ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
```

```
rtt min/avg/max/mdev = 193.653/229.499/299.873/49.766 ms
```

```
all ok, network is ready to use ... (end with CTRL-C)
```

steht die Verbindung und kann von der Kommandozeile

```
ping www.google.de
```

```
host www.google.de
```

oder im Internetbrowser benutzt werden.

## 6.Weitere Infos

[1] <http://askubuntu.com/questions/616245/tether-ubuntu-bq-phone>

[2] <http://ubuntufun.de/2015/06/so-nutzt-du-usb-tethering-mit-dem-ubuntu-phone/>

[3] [https://wiki.ubuntuusers.de/Baustelle/Ubuntu Touch Erweiterte Konfiguration/bq.txt](https://wiki.ubuntuusers.de/Baustelle/Ubuntu_Touch_Erweiterte_Konfiguration/bq.txt) (interessante Kommandos)

[4] <http://luki.org/2015/03/ubuntu-phone-erste-eindruecke/>

[5] <https://help.ubuntu.com/community/Router>

## Ubuntu-Handy als Internetgateway

zur Person:

Christoph Stockmayer ist seit 31 Jahren freiberuflicher Trainer auf den Gebieten Programmierung, C/C++/Java/Perl, OOA/OOD/UML und im gesamten UNIX/Linux-Sektor. Er ist SuSE Certified Linux-Trainer, LPIC1 und Lehrbeauftragter an der TU Nürnberg. Er betreut außerdem Programmier- und UNIX/Linux-Projekte.

