

MIT DEM NETBOOK INS NETZ

Mobilfunkner

Im Auslieferungszustand bringen gerade die kleinen Netbooks schlanke Linux-Betriebssysteme mit – knapp bemessener Festplattenplatz und das oft nur sieben Zoll kleine Display erfordern Anpassungen, die mit Windows XP oft kaum zu erreichen sind. Wie schlägt sich Linux beim Internetzugang per Bluetooth-Handy und UMTS-Stick?

VON MATTIAS SCHLENKER



Das Netbook als Bezeichnung für die Gerätegattung lässt keine Zweifel am Verwendungszweck der ultrabilligen Mini-Notebooks. Allerdings ist von Haus aus die Zugangsmöglichkeit auf WLAN beschränkt. Wer in einer Großstadt wohnt, in der jedes Café einen kostenlosen Hotspot für Latte-Macchiato-Konsumenten anbietet, den stört das nicht. Unterwegs oder auf dem Land ist WLAN dagegen nutzlos – um schnell einmal E-Mails zu prüfen, wird hier UMTS oder wenigstens GPRS benötigt.

Unser erster Versuchsaufbau umfasste den Eee PC im Auslieferungszustand, einen derzeit für zirka 15 Euro im Handel erhältlichen Micro-Bluetooth-Adapter, der fast vollständig im USB-Port verschwindet und ein schon etwas älteres Nokia 6021, das per Bluetooth angebunden als Modem funktionieren sollte. Für diesen Test verwendeten wir eine SIMYO-Karte, die bei GPRS-Verbindungen mit 24 Cent je MByte zu Buche schlägt, was akzeptabel für den gelegentlichen E-Mail-Check per IMAP ist.

Das Einstecken des Bluetooth-Dongles und der anschließende Blick in die Arbeitsflächen des Eee PCs fiel ernüchternd aus: Kein Programmsymbol *Bluetooth* und keine Möglichkeit, die Internetverbindung per Modem einzurichten. Handarbeit ist angesagt: Mit *Strg-Alt-T* öffnen Sie ein Terminalfenster, in diesem wiederum erlangen Sie mit dem Befehl *sudo su* die Administratorrechte, die für die folgenden Schritte benötigt werden. Als Erstes steht der Start des DBus-Daemons an, dann der Start des eigentlichen Bluetooth-Services:

```
/etc/init.d/dbus start
/etc/init.d/bluetooth start
```

Nun kann ermittelt werden, ob der Bluetooth-Adapter richtig erkannt und aktiviert wurde. Das erledigt der Befehl

```
hciconfig -a
```

Neben der nicht leeren BD-Adresse sollte ein zufällig generierter Name Ihres Netbooks angezeigt werden. Die komplette Ausgabe zeigt unser Screenshot. Ist diese Hürde gemeistert, steht die Suche nach dem Mobiltelefon an. Auch diese ist auf der Kommandozeile vorzunehmen:

```
hcitool scan
```

In der Ausgabe steht neben dem Namen des Telefons dessen Bluetooth-Adresse. Die sollten Sie gut notieren. Es folgt die Anpassung von drei Konfigurationsdateien, was Sie am besten mit dem Editor *kwrit* vornehmen, den Sie in der eben offenen Konsole gefolgt vom Dateinamen starten. Zuerst ist */etc/bluetooth/hcid.conf* an der Reihe. Hier ändern Sie die Zeile

```
user@asus-1864595581
~/home/user> sudo su
user@asus-1864595581:~$ /etc/init.d/dbus start
Starting system message bus: dbus.
user@asus-1864595581:~$ /etc/init.d/bluetooth start
Starting bluetooth: hcid sdpd sdp_options rfcomm.
user@asus-1864595581:~$ hciconfig -a
hci0:   Type: USB
        BD Address: 00:1B:3C:00:42:0F ACL MTU: 300:10 SCO MTU: 64:8
        LMP: RUNNING PSCO:
        RX bytes:1745 acl:0 sco:0 events:29 errors:0
        TX bytes:2630 acl:0 sco:0 commands:28 errors:0
        Features: 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        Packet type: DFL DMS DMS DFL DMS DMS DMS DMS
        Link policy: RSWITCH HOLD SNIFF FWAC
        Link mode: SLAVE ACCEPT
        Name: 'asus-1864595581-0'
        Class: 0x3A0100
        Service Classes: Networking, Rendering, Capturing, Object Transfer, Audio
        Device Class: Computer, Unspecified
        HCI Ver: 2.0 (0x3) HCI Rev: 0x2c LMP Ver: 2.0 (0x3) LMP Subver: 0x0c
        Manufacturer: Cambridge Silicon Radio (10)
```

Ist Bluetooth gestartet, können via Kommandozeile der Adapter sowie in der Umgebung befindliche Geräte ermittelt werden.

```
security auto;
```

```
auf
```

```
security user;
```

und tragen als *passkey* einen neuen PIN-Code für eingehende Verbindungen ein – sonst könnte jeder Fremde mit „1234“ eine Verbindung aufbauen. Es folgt die Datei */etc/bluetooth/rfcomm.conf*, in der Sie Bluetooth-Adresse, Kanal und eine kurze Beschreibung eingeben:

```
rfcomm0 {
    bind yes;
    device 00:13:70:00:53:F1;
    channel 1;
    comment „Nokia 6021 als
    Modem“;
}
```

Nun folgt die Datei */var/lib/bluetooth/00:00:00:00:00:00/pincodes* (statt lauter Nullen ist die BD-Adresse Ihres Bluetooth-Dongles zu verwenden, beim Aufruf auf der Kommandozeile ist der Doppelpunkt durch einen Backslash zu maskieren). In diese Datei tragen Sie die Bluetooth-Adresse des Mobiltelefons und die PIN für ausgehende Verbindungen ein (hier „4567“, diese wird später auf dem Handy abgefragt):

```
00:13:70:00:53:F1 4567
```

Sollten Sie mehrere Handys mit dem Eee PC nutzen wollen, verwenden Sie eine Zeile für jedes Gerät.

Paarungsbereit

Es folgen der Neustart des Bluetooth-Subsystems und ein kleiner Test:

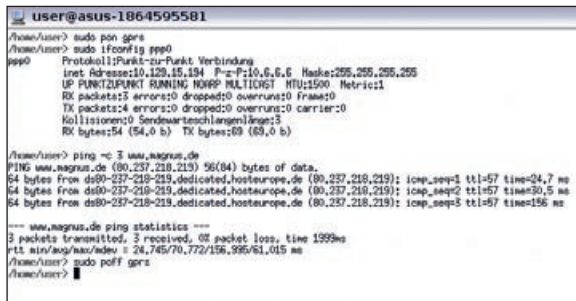
```
/etc/init.d/bluetooth stop
/etc/init.d/bluetooth start
cat < /dev/rfcomm0
```

Wird jetzt auf dem Handy das Bluetooth-Passwort abgefragt und anschließend eine Verbindung aufgebaut, können Sie mit *Strg-C* die Konsole wieder freigeben. Unser Ausflug sollte nun eigentlich beendet sein: In der Theorie sollte sich der Aufbau der Verbindung mit dem *Xandros-Verbindungsassistent* herstellen lassen. In der Praxis scheiterte dies an der nicht erkannten Schnittstelle */dev/rfcomm0*. Wir begaben uns daher erneut auf die Kommandozeile und erstellten ein Chat-Script */etc/chatscripts/gprs* (der Name kann beinahe willkürlich gewählt werden, Sie können das Script auch nach Ihrem Provider benennen):

```
TIMEOUT      5
ECHO         ON
ABORT        , \nBUSY\r'
ABORT        , \nERROR\r'
ABORT        , \nNO ANSWER\r'
ABORT        , \nNO CARRIER\r'
ABORT        , \nNO DIALTONE\r'
ABORT        , \nRINGING\r\n\r\nRING
ING\r'
, '
\ rAT
TIMEOUT 15
OK         ATE1
OK         ,AT+CGDCONT=1,"IP","in
ternet.eplus.de","0,0'
OK         ATD*99***1#
CONNECT    ""
```

In der Regel werden Sie nur den Access-Point-Namen (hier *internet.eplus.de*) anpassen müssen (beispielsweise *web.vodafone.de* für Vodafone oder *smartsites.t-mobile* bei den T-Mobile Web'n'Walk-Tarifen). Da der APN-Name im schlimmsten Fall den Unterschied zwischen Flatrate und volumenbasiertem Tarif ausmachen kann, sollten Sie den APN-Namen auf den Webseiten des Providers nachschlagen. Die Einwahlnummer **99***1#* sollte überall gleich sein. Ist das Chat-Script erstellt, folgt die *Peer-Konfiguration*, in unserem Beispiel unter */etc/ppp/peers/gprs*:

```
/dev/rfcomm0 115200
connect ,/usr/sbin/chat -v -f /etc/chat
scripts/gprs'
noauth
defaultroute
usepeerdns
debug
noipdefault
local
icpc-accept-local
novj
novjccomp
```



Nach der Anpassung von drei Konfigurationsdateien genügt ein *sudo pon gprs* für die Internetwahl per Bluetooth.

```
lcp-echo-failure 10000
lcp-echo-interval 1000
```

Die hier vorgenommenen Einstellungen sind eher geräte- als verbindungsspezifisch. Je nach Handy und späterem Verbindungstyp sind Experimente mit schnelleren Datenraten sinnvoll, bei vielen UMTS-Geräten dürfte 864000 angemessen sein.

Eine Verbindung bauen Sie nun mit

```
pon gprs
```

auf, bzw. *pon* gefolgt vom Namen Ihrer Peer-Konfiguration. Während des Verbindungsaufbaus sollte wenig mehr passieren, als dass Ihr Handy wegen der Bluetooth-Verbindung nachfragt und anschließend das GPRS- oder 3G-Symbol im Display zeigt. Dass eine Ver-

Nützliche Weblinks

- <http://oerks.de/tech/blueZ>
Alexander Stielau erklärt die Bluetooth-Konfiguration und -Fehlersuche bei Debian-Derivaten wie dem von Asus verwendeten Xandros
- https://forge.betavine.net/frs/?group_id=12&release_id=116
Ein Linux-Konfigurationstool für verschiedene UMTS-Karten ist auf Vodafones Entwicklerseite erhältlich
- <http://umtsmon.sf.net>
UMTSmon ist ein freies Tool zur Verwaltung von UMTS-Karten. Es zeigt ausführliche Verbindungsstatistiken und arbeitet auch mit einigen Mobiltelefonen zusammen.
- <http://blog.rootserverexperiment.de/category/eeepc/>
Ubuntu-Images für den Eee PC bietet der Autor auf seiner Webseite; diese verfügen über eine bessere Hardware-Unterstützung als das original installierte Xandros

bindung besteht, zeigt der Befehl

```
ifconfig ppp0
```

Jetzt können Sie etwas im Internet surfen und E-Mails überprüfen. Zum Trennen der Verbindung nutzen Sie einfach den Befehl *pon off gprs*. Auch diesen Vorgang sollte das Mobiltelefon anzeigen und *ifconfig* bestätigen. Der Weg bis hierhin war relativ steinig, da jedoch alle komplexen Einstellungen in dauerhaft gespeicherten Konfigurationsdateien stehen, genügt nach

einem Neustart der Start der Dienste *dbus* und *bluetooth* gefolgt von *pon gprs*, um im Netz zu sein. Das lässt sich weiter verkürzen, indem Sie eine Datei */etc/opt/xandros/xandrosncs/hooks/post-up.d/99bluetooth* anlegen, welche lediglich die folgenden drei Zeilen enthält:

```
#!/bin/sh
/etc/init.d/dbus restart
/etc/init.d/bluetooth restart
```

Verwenden Sie nach dem Speichern den Befehl

```
chmod 0755 /etc/opt/xandros/xandrosncs/
hooks/post-up.d/99bluetooth
```

um Ausführungsrechte zu setzen. Nach dem Neustart genügt es, mit *Strg-Alt-T* ein Terminal zu öffnen und in diesem *sudo pon gprs* einzugeben, um mit dem Internet zu verbinden, *sudo pon off gprs* trennt die Verbindung wieder.

Und UMTS?

Glücklich kann sich schätzen, wer ein UMTS-Handy mit Bluetooth-Schnittstelle besitzt: Die Konfiguration entspricht exakt der oben beschriebenen. Lediglich Verbindungsgeschwindigkeit und Name des Access Points müssen noch angepasst werden. In der Regel verwendet das UMTS-Handy bei fehlender 3G-Verbindung als Fallback das langsamere GPRS-Protokoll.

Allerdings verliert die Kombination aus Handy und Bluetooth-Stick derzeit an Popularität: Separate Nur-Datenverträge sind oft günstiger als zubuchbare Datentarife. Statt eines UMTS-Handys kommt hier ein einfacher USB-Stick zum Einsatz, der die SIM-Karte aufnimmt.

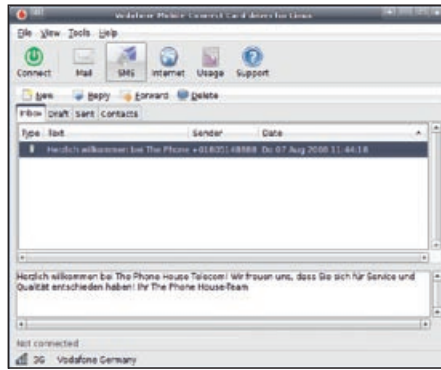
Für unsere Tests stand ein *Huawei*-Stick *E172* mit *Vodafone*-Branding zur Verfügung, der von halbwegs aktuellen Linux-Kerneln als USB-Modem erkannt wird. Tests mit anderen Sticks von *Huawei* und *Option* zeigten das gleiche Bild: Nach dem Anstecken des Sticks wurden zwei oder mehr serielle Schnittstel-

len `/dev/ttyUSB0` etc. angelegt. Die Ausgabe von `lsusb` und `dmesg` zeigt bei einigen Sticks neben den seriellen Schnittstellen mit der Benennung *airprime converter* noch einen USB-Massenspeicher. Es handelt sich dabei um eine kleine, mit ISO9660-Dateisystem versehene Flash-Partition, auf der sich die Windows-Treiber des Sticks befinden. Dieses virtuelle CD-Laufwerk führt bei einigen Kernel-Stick-Kombinationen zu Problemen bei der Initialisierung der seriellen Schnittstelle des UMTS-Modems. Sollte nur das CD-Laufwerk gefunden werden, entfernen Sie den Stick, entladen und blacklisten Sie das Modul `usb_storage` und laden Sie dieses gegebenenfalls nach dem Anstecken des UMTS-Sticks wieder nach.

Übrigens sollten UMTS-Karten am Cardbus oder Express-Card-Steckplatz auf die gleiche Weise erkannt werden: Derartige Karten enthalten meist die gleichen Chipsätze wie die USB-Pendants und binden diese über eine interne USB-Schnittstelle an. Werden keine seriellen Schnittstellen erkannt, sollten Sie sich nicht allzu lange mit gepatchten Modulen aufhalten: Installieren Sie in diesem Fall einen aktuellen Kernel oder wechseln Sie zur neuesten Version Ihrer Distribution.

AT-Befehle: Alte Bekannte

Die Ansteuerung des UMTS-Modems erfolgt auf einer der ersten drei seriellen Schnittstellen mit gewöhnlichen AT-Befehlen. Theoretisch würde also nichts dagegen sprechen, Chat-Skripte für die UMTS-Einwahl zu nutzen. Gegen diesen Weg sprechen aber einige kleine Teufel im Detail: Weil die UMTS-Sticks weder über Display, noch über Eingabemöglichkeiten verfügen, muss die gesamte Kommunikation über AT-Befehle erfolgen. So auch die Übergabe der SIM-PIN, die nur einmal zwischen dem Einstecken des Sticks und der ersten Einwahl erfolgen darf. Weil



Vorbildlich: Dank zunehmender Linux-Verbreitung auf Netbooks arbeitet Vodafone an einem eigenen Konfigurations- und Zugangstool.

die Auswertung der Rückgabe-Parameter zu aufwendig für herkömmliche Chat-Skripte ist, raten wir zur Verwendung dezidiert UMTS-Einwahlprogramme. Dies ist auch der Grund, warum einfache Einwahlprogramme wie das im Xandros des Eee PC enthaltene oft nicht funktionieren.

Derzeit stehen zwei umfangreiche Einwahlprogramme bereit, die eine Vielzahl an Karten kennen. Eines ist das von einer freien Entwicklergemeinde betreute *Umtsmon*, das zweite der maßgeblich von Vodafone vorangetriebene *Vodafone Mobile Connect Card Driver for Linux* (VMC). VMC steht ebenfalls unter GPL und ist komplett im Quellcode verfügbar. Für Netbooks, die unter Ubuntu oder anderen nicht Netbook-spezifischen 32-Bit Distributionen laufen, stellt Vodafone einen All-In-One-Installer bereit, der Python und alle benötigten Werkzeuge mitbringt. Das macht das Programm zwar fett, stellt aber die Lauffähigkeit auf einer Vielzahl an Distributionen sicher.

Nach dem Herunterladen der aktuellen Version, werden die Zugriffsrechte zunächst auf „ausführbar“ gesetzt, dann starten Sie die Installation mit vorangestelltem `sudo`:

```
chmod a+x vodafone-mobile-connect-
card-driver-for-linux-2.0.beta3-
ALL-i386-installer.run
sudo ./vodafone-mobile-connect-card-
driver-for-linux-2.0.beta3-ALL-i386-
installer.run
```

Nach abgeschlossener Installation sollten Sie einen Menüeintrag vorfinden, welcher VMC startet. Auf unserem Testrechner schlug der Start mit Ubuntu 8.04.1 jedoch fehl, gelang aber in einer Konsole mit

```
sudo vodafone-mobile-connect-card-
driver-for-linux
```

Nach korrekter Eingabe von PIN und Namen des APN `event.vodafone.de` - unser Testmodem war mit einer Prepaid-Vodafone-SIM des Websession-Paketes ausgestattet - gelang die Einwahl sofort ohne Probleme. Um nicht immer die Konsole bemühen zu müssen, änderten wir den Menüeintrag für VMC mit vorangestelltem `sudo` ab und vergaben per `visudo` Verbindungsrechte für eine bestimmte Nutzergruppe:

```
User_Alias UMTS = mattias, anja, oma
UMTS ALL=NOPASSWD: /usr/bin/vodafone-
mobile-connect-card-driver-for-linux
```

Fazit

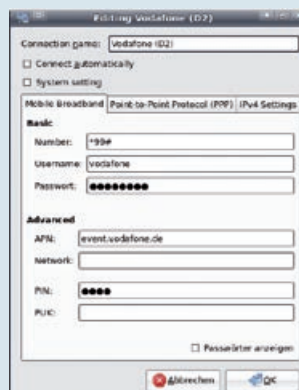
Dank der Verwendung bewährter Standards und simpler Hardware ist die UMTS-Einwahl unter Linux kein Hexenwerk und geht in vielen Fällen auch ohne die Anpassung von Chat-Skripten vonstatten. Allerdings werden wohl noch einige Monate vergehen, bis die Integration der Einwahlprogramme ins Betriebssystem soweit fortgeschritten ist, dass der Komfort eines WLAN-Beitritts erreicht wird. Der neue *Network Manager* in Ubuntu 8.10 macht einen wichtigen Schritt in diese Richtung. jkn

Alles unter einem Dach

Der neue *Network Manager* von Ubuntu 8.10 bringt einen UMTS-Verbindungsassistenten gleich mit. Damit kann blitzschnell zwischen den Verbindungsarten gewechselt werden. Steckt man den *Huawei E172* an ein System mit Ubuntu 8.10 und laufendem *Network Manager* an, erkennt dieser sofort das UMTS-Modem und bietet an, eine Verbindung zu konfigurieren. Die Eingabefelder in diesem Dialog genügen jedoch in vielen Fällen nicht - die meisten Provider erfordern einen anderen APN sowie die Angabe einer PIN. Um diese einzu-

Keine Zukunftsmusik: Der neue *Network Manager* bietet eine simple Einwahlmöglichkeit, liefert allerdings keine Rückmeldungen zur Verbindungsqualität.

tragen, klicken Sie mit der rechten Maustaste auf das Netzwerksymbol im Tray und wählen *Edit Connections*. Im Reiter *Mobile Broadband* können Sie



dann die Verbindung auswählen und mit *Bearbeiten* die fehlenden Parameter hinzufügen.

Leider unterstützt der *Network Manager* derzeit nur USB-(sowie Cardbus- und ExpressCard-)Modems, erkennt die SIM noch nicht selbstständig und erlaubt keine explizite Anwahl einer bestimmten Hardware.