

Come sfruttare in Linux la piena capacità dei modem HSDPA

Inviato da PoG
 martedì 23 ottobre 2007
 Ultimo aggiornamento giovedì 24 gennaio 2008

Questa guida raccoglie le informazioni necessarie per configurare i modem HSDPA in modo da sfruttare la piena capacità offerta, superando le limitazioni di velocità di 64 KB/s imposte dal modulo usbserial che normalmente si usa per configurare tali modem sotto Linux.

Come sfruttare in Linux la piena capacità dei modem HSDPA

I modem HSDPA sono dispositivi, tipicamente USB, che consentono l'accesso ad internet mediante una rete GSM/GPRS/UMTS/HSDPA con velocità di connessioni teoriche pari a 3,6 Mbps.

Il modo più semplice di configurarli in ambiente Linux è quello di utilizzare il modulo usbserial, come descritto ampiamente in articoli presenti in questo sito e a cui si rimanda per le configurazioni di dettaglio (rif. per il modem ONDA M1HS e per il modem Huawei E220).

Purtroppo il modulo usbserial sfrutta un buffer di piccole dimensioni che limita la velocità di connessione in download a 64 KB/s.

Per rimuovere tale limitazione è possibile seguire essenzialmente due strade:

Modificare il modulo usbserial in modo da introdurre un parametro che consenta di impostare l'ampiezza massima del buffer utilizzato;

- Modificare il modulo airprime, più evoluto di usbserial, per fare riconoscere il modem HSDPA e sfruttarne la piena capacità.

Nel seguito riporto le prove da me effettuate utilizzando le distribuzioni openSuse 10.2/3 e (K)ubuntu 7.04 Feisty Fawn.

Per implementare le modifiche esposte è necessario avere installato i sorgenti del kernel e il compilatore gcc ed eseguire i comandi in una shell di root.

Configurazione in openSuse 10.2

Preliminarmente, se non l'avete già fatto, installate il compilatore gcc e i sorgenti del kernel mediante YaST.

Il modulo airprime presente in openSuse 10.2 non è predisposto per l'utilizzo di buffer che consentono connessioni più veloci di 64 KB/s. Per tale motivo conviene modificare il driver usbserial per introdurre un nuovo parametro che consente di fissare la dimensione massima del buffer.

A tal fine è necessario applicare una patch al modulo usb-serial.c presente in:

`/usr/src/linux/drivers/usb/serial` Preliminarmente scaricate dalla memoria il modulo usbserial con il comando:

`modprobe -r usbserial` La patch in questione è di seguito riportata:

```
--- usb-serial.c.orig 2007-09-12 00:25:50.000000000 +0200
+++ usb-serial.c 2007-09-12 00:29:57.000000000 +0200
@@ -58,6 +58,7 @@
 */
static int debug;
+static ushort maxSize = 0;
static struct usb_serial *serial_table[SERIAL_TTY_MINORS]; /* initially all NULL */
static LIST_HEAD(usb_serial_driver_list);
@@ -816,6 +817,7 @@
goto probe_error;
}
buffer_size = le16_to_cpu(endpoint->wMaxPacketSize);
+ buffer_size = (endpoint->wMaxPacketSize > maxSize)?endpoint->wMaxPacketSize:maxSize;
port->bulk_in_size = buffer_size;
port->bulk_in_endpointAddress = endpoint->bEndpointAddress;
port->bulk_in_buffer = kmalloc (buffer_size, GFP_KERNEL);
@@ -1190,3 +1192,5 @@
module_param(debug, bool, S_IRUGO | S_IWUSR);
MODULE_PARM_DESC(debug, "Debug enabled or not");
+module_param(maxSize, ushort, 0);
+MODULE_PARM_DESC(maxSize, "User specified USB endpoint size");
```

Per applicarla seguite i passi descritti:

- Copiare le righe di testo della patch e incollarle in un nuovo documento di testo (utilizzando il vostro editor preferito, come kwrite o kate)
 - Salvate il file come usb-serial-patch.c in una directory a vostra scelta (ad esempio /home/<utente>/patch)
 - Spostarsi nella directory del modulo usb-serial.c
cd /usr/src/linux/drivers/usb/serial
 - Fare una copia di backup del modulo usb-serial.c originale:
cp usb-serial.c usb-serial.c.bak
 - Applicare la patch con il comando:
patch -n0 /percorso/della/patch/usb-serial.patch.c usb-serial.c
 - Compilare i moduli presenti nella directory:
make -C /lib/modules/`uname -r`/build M=`pwd`
 - Al termine della compilazione digitare il comando di aggiornamento di dipendenza dei moduli:
depmod -a
 - Copiare il modulo creato usb-serial.ko nella directory dei moduli appropriata:
cp usb-serial.ko /lib/modules/`uname -r`/kernel/drivers/usb/serial/
 - Installare il nuovo modulo usbserial passando un ulteriore parametro al comando modprobe che consente di modificare la grandezza del buffer:
modprobe vendor=0x05c6 product=0x6613 maxSize=4096
 - Verificare la creazione dei device /dev/ttyUSBn (n=0, 1, 2)
- Il parametro maxSize nell'esempio è impostato a 4096, ma nulla vieta di impostarlo ad un valore maggiore. Io, ad esempio, uso un valore di 8192.

Configurazione in openSuse 10.3

La configurazione del driver in openSuse 10.3 è semplificata dal fatto che il modulo airprime è già predisposto per superare le limitazioni a 64 KB/s della velocità di collegamento a Internet.

Di seguito riporto la parte iniziale del sorgente airprime.c:

```
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/tty.h>
#include <linux/tty_flip.h>
#include <linux/module.h>
#include <linux/usb.h>
#include <linux/usb/serial.h>
static struct usb_device_id id_table [] = {
{ USB_DEVICE(0x0c88, 0x17da) }, /* Kyocera Wireless KPC650/Passport */
{ },
};
MODULE_DEVICE_TABLE(usb, id_table);
#define URB_TRANSFER_BUFFER_SIZE 4096
#define NUM_READ_URBS 4
#define NUM_WRITE_URBS 4
#define NUM_BULK_EPS 3
#define MAX_BULK_EPS 6
```

Dal codice si vede chiaramente il parametro utilizzato per stabilire l'ampiezza del buffer (URB_TRANSFER_BUFFER_SIZE) e la parte relativa al riconoscimento dell'hardware (USB_DEVICE...).

Occorre pertanto modificare il modulo in questione per consentire il riconoscimento del modem HSDPA in nostro possesso. Nel mio caso, il modem (ONDA M1HS) ha idVendor e idProduct pari a 0x05c6 e 0x6613 rispettivamente e ho modificato il modulo aggiungendo una riga USB_DEVICE con i parametri del modem:

```
static struct usb_device_id id_table [] = {
{ USB_DEVICE(0x0c88, 0x17da) }, /* Kyocera Wireless KPC650/Passport */
{ USB_DEVICE(0x05c6, 0x6613) }, /* ONDA M1HS */
{ USB_DEVICE(0x12d1, 0x1003) }, /* Huawei E220 */
{ },
};
```

A questo punto basta ricompilare il modulo, aprire una shell come root ed eseguire questi comandi:

```
cd /usr/src/linux/drivers/usb/serial/
make -C /lib/modules/`uname -r`/build M=`pwd` aggiornare la dipendenza dei moduli configurarli:
depmod -ae copiare il modulo appena compilato nella directory dei moduli appropriata:
```

cp airprime.ko /lib/modules/`uname -r`/kernel/drivers/usb/serial/Se tutto è andato per il verso giusto, è possibile configurare il modem mediante il comando:
 modprobe airprimeVerificate la presenza dei dispositivi /dev/ttyUSBn (n=0,1, ...); la porta seriale da utilizzare per la configurazione dei client ppp è sempre la ttyUSB0.

Non è necessario inserire nessun comando nel file /etc/init.d/boot.local in quanto il driver airprime viene correttamente caricato ad ogni avvio del sistema.

Configurazione in (K)Ubuntu Feisty Fawn 7.04

Anche in Ubuntu il driver airprime è predisposto per l'utilizzo di buffer che consentono di superare i limiti di banda di 64 KB/s. Occorre solamente aggiungere al driver i parametri del modem in nostro possesso e ricompilare il modulo, come descritto per la openSuse 10.3.

L'unica complicazione in tali distribuzioni è dettata dal fatto che i sorgenti del kernel non sono disponibili nel CD di installazione e occorre scaricarli da Internet.

Per far ciò configurate il modem HSDPA in modo tradizionale usando il driver usbserial, scaricate e installate i pacchetti build-essential, linux-header e linux-source mediante synaptic o, se preferite, mediante la riga di comando (dato come root)

apt-get install build-essential linux-header linux-sourceDecomprimate i sorgenti del kernel mediante i comandi:

```
cd /usr/src
```

```
tar xjvf linux-source-<versione>.tar.bz2
```

Note finali

Con i metodi descritti si supera il limite di 64 KB/s imposto dall'uso di usbserial nei collegamenti ad Internet mediante modem HSDPA.

Le velocità effettive di collegamento dipendono ovviamente dal carico della rete dell'operatore; nelle mie prove ottengo velocità di connessione tipiche di 220-250 KB/s.

Quando ho configurato il driver airprime di Ubuntu ho eseguito un upgrade della distribuzione ottenendo velocità pari a 280 KB/s con punte di 370 KB/s.

Il problema dell'utilizzo di un modulo personalizzato (usbserial o airprime) è che si perdono le modifiche effettuate ai driver ad ogni aggiornamento del kernel della distribuzione.

Per rendere meno noioso il processo di riconfigurazione è possibile ricavarci le patch dei moduli e racchiudere tutte le operazioni in un piccolo script di shell. Lo script può essere quello di seguito riportato per i due casi esaminati

Script per usbserial

```
#!/bin/bash
```

```
/sbin/modprobe -r usbserial
```

```
cd /usr/src/linux/drivers/usb/serial/
```

```
cp usb-serial.c usb-serial.bak.c
```

```
/usr/bin/patch -p0 usb-serial.c /root/patches/usb-serial-patch.c
```

```
make -C /lib/modules/`uname -r`/build M=`pwd`
```

```
cp usb-serial.ko /lib/modules/`uname -r`/kernel/drivers/usb/serial
```

```
depmod -a
```

```
/sbin/modprobe usbserial vendor=0x05c6 product=0x6613 maxSize=8192
```

```
ls/dev/ttyUSB*
```

```
echo "Patch applicata. Configurazione terminata"
```

Script per airprime

```
#!/bin/bash
```

```
/sbin/modprobe -r airprime
```

```
cd /usr/src/linux/drivers/usb/serial/
```

```
cp airprime.c airprime.bak.c
```

```
/usr/bin/patch -p0 airprime.c /root/patches/airprime-patch.c
```

```
make -C /lib/modules/`uname -r`/build M=`pwd`
```

```
cp airprime.ko /lib/modules/`uname -r`/kernel/drivers/usb/serial
```

```
depmod -a
```

```
/sbin/modprobe airprime
```

```
ls/dev/ttyUSB*
```

```
echo "Patch applicata. Configurazione terminata"
```

Negli script riportati si assume che le patch risiedono nella directory /root/patches.

Fonti

<http://samat.org/weblog/20070127-high-speed-cellular-wireless...>

http://www.junxion.com/opensource/linux_highspeed_usbserial.html

<http://mybroadband.co.za/vb/showthread.php?t=71517>