

Using an Iridium phone with GNU/Linux and PPP

Please [inform the author](#) of any mistakes.

Iridium, windows, GNU/Linux

The [Iridium satellite system](#) officially supports data connections only with proprietary Windows software. This shuts out anyone who needs a reliable computer system ([our application](#) is for a remote, un-personned, site on the Antarctic plateau, so Windows is completely unsuitable). Fortunately, the Iridium system does work with standard PPP software, making it accessible to GNU/Linux, although there are no guarantees that this situation will continue.



Varieties of Iridium phone/modem

The common varieties of Iridium phone are the Motorola 9500 and 9505. The phone contains a fairly standard modem that responds to most of the [usual AT commands and a few extra ones](#) (see the [Bibliography](#) below for a complete list of relevant documents on this web site), with [some caveats](#), and is accessible via an RS-232 serial link at 19200 baud (note that you will need the data connector on the base of the phone in order to attach a 9-pin D connector). It is also possible to buy an Iridium modem, which is basically the internals of the 9505.

Varieties of data connection

There are [two ways of making an outgoing data connection with the Iridium system](#): either dial your own ISP and make a data connection over the normal phone system, or use the Iridium gateway as your ISP (so-called "direct Internet data"). This latter mode is presumably a better way to connect to the Internet.

The baud rate of the data connection is nominally 2400 (apparently [it can vary from 2200 to 3800 baud](#) depending on antenna efficiency and atmospheric conditions). The Windows software claims 10kbps using some sort of proprietary data compression, but this obviously only applies to data that can be compressed. Under GNU/Linux you can use, e.g., "ssh -C" to make a ssh connection with compression.

Instructions for establishing a direct Internet connection

The following instructions apply to a Motorola 9505 and Redhat 7.3, and are for direct Internet connection via the Iridium gateway. Note that US DoD users have a special gateway in Hawaii, which may or may not work with these instructions.

Begin by using the standard Redhat tool to add a new network connection via PPP. The modem should be set to 19200 baud, 8 data bits, 1 stop bit, no parity, hardware handshaking. After you have finished, edit your /etc/wvdial.conf and make it look like this:

```
[Modem0]
Modem = /dev/ttyS0
Baud = 19200
SetVolume = 2
Dial Command = ATDT
Init1 = ATZ
Init2 = AT+M0=1V1X4E1Q0&c1
Init3 = AT+cbst=71,0,1
Init4 = AT+cr=1
FlowControl = CRTSCTS
```

```
[Dialer Iridium gateway]
Phone = 008816000023
Password = none
Username = none
Stupid Mode = 1
Inherits = Modem0
```

The interesting things about this file are (1) the baud rate of 19200 is just for serial communication between the computer and the phone, and is matched to the capabilities of the modem, the outgoing baud rate is still 2400, (2) the "AT+cbst=71,0,1" line is crucial, (3) the 008816000023 is an undocumented Iridium gateway number that allows you to connect without having to specify a password and username (this speeds up the connection). An alternative is to use 008816000022, in which case you have two options: Password is "password" and Username is your Iridium telephone number, or, Password and Username are both "directinternet".

You can now activate the Iridium connection using "ifup ppp0". You should see the Iridium phone dialling out, and after about 28 seconds you should be connected. If you have PPP configured with debugging (by putting the line "debug" in /etc/ppp/options or /etc/ppp/peers/wvdial (I'm not sure which is best)), you should see something like the following in /var/log/ppplog:

```
Nov 25 10:42:06 mcba1 pppd[24727]: pppd 2.4.1 started by root, uid 0
Nov 25 10:42:06 mcba1 ifup-ppp: pppd started for ppp0 on /dev/ttyS0 at 19200
Nov 25 10:42:06 mcba1 WvDial: WvDial: Internet dialer version 1.41
Nov 25 10:42:06 mcba1 WvDial: Initializing modem.
Nov 25 10:42:06 mcba1 WvDial: Sending: ATZ
Nov 25 10:42:06 mcba1 WvDial: ATZ
Nov 25 10:42:06 mcba1 WvDial: OK
Nov 25 10:42:06 mcba1 WvDial: Sending: ATS0=1V1X4E1Q0&c1
Nov 25 10:42:06 mcba1 WvDial: ATS0=1V1X4E1Q0&c1
Nov 25 10:42:06 mcba1 WvDial: OK
Nov 25 10:42:06 mcba1 WvDial: Sending: AT+cbst=71,0,1
Nov 25 10:42:06 mcba1 WvDial: AT+cbst=71,0,1
Nov 25 10:42:06 mcba1 WvDial: OK
Nov 25 10:42:06 mcba1 WvDial: Sending: AT+cr=1
Nov 25 10:42:06 mcba1 WvDial: AT+cr=1
Nov 25 10:42:06 mcba1 WvDial: OK
Nov 25 10:42:06 mcba1 WvDial: Modem initialized.
Nov 25 10:42:06 mcba1 WvDial: Sending: ATDT 008816000023
Nov 25 10:42:06 mcba1 WvDial: Waiting for carrier.
Nov 25 10:42:06 mcba1 WvDial: ATDT 008816000023
Nov 25 10:42:22 mcba1 WvDial: +CR: REL ASYNC
Nov 25 10:42:22 mcba1 WvDial: CONNECT 19200
Nov 25 10:42:22 mcba1 WvDial: Carrier detected. Starting PPP immediately.
Nov 25 10:42:22 mcba1 pppd[24727]: Serial connection established.
Nov 25 10:42:22 mcba1 pppd[24727]: using channel 5
Nov 25 10:42:22 mcba1 pppd[24727]: Using interface ppp0
Nov 25 10:42:22 mcba1 pppd[24727]: Connect: ppp0 <--> /dev/ttyS0
Nov 25 10:42:23 mcba1 pppd[24727]: sent [LCP ConfReq id=0x1 <asynmap 0x0> <magic 0x
Nov 25 10:42:24 mcba1 pppd[24727]: rcvd [LCP ConfReq id=0x1 <asynmap 0xa0000> <magi
Nov 25 10:42:24 mcba1 pppd[24727]: sent [LCP ConfAck id=0x1 <asynmap 0xa0000> <magi
Nov 25 10:42:24 mcba1 pppd[24727]: rcvd [LCP ConfAck id=0x1 <asynmap 0x0> <magic 0x
Nov 25 10:42:24 mcba1 pppd[24727]: sent [IPCP ConfReq id=0x1 <addr 166.84.159.111> <
Nov 25 10:42:24 mcba1 modprobe: modprobe: Can't locate module ppp-compress-21
Nov 25 10:42:26 mcba1 pppd[24727]: rcvd [IPCP ConfReq id=0x1 <addr 172.16.1.1>]
Nov 25 10:42:26 mcba1 pppd[24727]: sent [IPCP ConfAck id=0x1 <addr 172.16.1.1>]
Nov 25 10:42:26 mcba1 pppd[24727]: rcvd [CCP ConfReq id=0x1 < 11 05 00 01 03> < 12 0
Nov 25 10:42:26 mcba1 modprobe: modprobe: Can't locate module ppp-compress-21
Nov 25 10:42:26 mcba1 pppd[24727]: sent [CCP ConfReq id=0x1]
Nov 25 10:42:26 mcba1 pppd[24727]: sent [CCP ConfRej id=0x1 < 11 05 00 01 03> < 12 0
Nov 25 10:42:26 mcba1 pppd[24727]: rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
Nov 25 10:42:26 mcba1 pppd[24727]: sent [IPCP ConfReq id=0x2 <addr 129.94.162.191> <
Nov 25 10:42:29 mcba1 pppd[24727]: sent [CCP ConfReq id=0x1]
```

```

Nov 25 10:42:29 mcba1 pppd[24727]: sent [IPCP ConfReq id=0x2 <addr 129.94.162.191> <
Nov 25 10:42:30 mcba1 pppd[24727]: rcvd [CCP ConfAck id=0x1]
Nov 25 10:42:30 mcba1 pppd[24727]: rcvd [CCP ConfReq id=0x2]
Nov 25 10:42:30 mcba1 pppd[24727]: sent [CCP ConfAck id=0x2]
Nov 25 10:42:30 mcba1 pppd[24727]: rcvd [IPCP ConfNak id=0x2 <addr 172.16.8.247> <ms
Nov 25 10:42:30 mcba1 pppd[24727]: sent [IPCP ConfReq id=0x3 <addr 172.16.8.247> <ms
Nov 25 10:42:32 mcba1 pppd[24727]: rcvd [CCP ConfAck id=0x1]
Nov 25 10:42:33 mcba1 pppd[24727]: rcvd [IPCP ConfNak id=0x2 <addr 172.16.8.247> <ms
Nov 25 10:42:33 mcba1 pppd[24727]: rcvd [IPCP ConfAck id=0x3 <addr 172.16.8.247> <ms
Nov 25 10:42:33 mcba1 pppd[24727]: local IP address 172.16.8.247
Nov 25 10:42:33 mcba1 pppd[24727]: remote IP address 172.16.1.1
Nov 25 10:42:33 mcba1 pppd[24727]: primary DNS address 204.117.214.10
Nov 25 10:42:33 mcba1 pppd[24727]: secondary DNS address 199.2.252.10
Nov 25 10:42:33 mcba1 pppd[24727]: Script /etc/ppp/ip-up started (pid 24767)
Nov 25 10:42:33 mcba1 pppd[24727]: Script /etc/ppp/ip-up finished (pid 24767), statu
Nov 25 10:42:33 mcba1 pppd[24727]: Script ?? finished (pid 24754), status = 0x0

```

This shows that the Iridium gateway is at address 172.16.1.1, and it has assigned the PC the IP number 172.16.8.247. Both of these numbers are so-called RFC1597 class B numbers reserved for private networks (i.e., they are not accessible directly via the Internet). The Iridium gateway handles the routing of packets from the private network to the rest of the Internet. Domain Name Servers have also been specified by Iridium as part of the PPP initialisation process, and you should see these appearing in your /etc/resolve.conf file.

Your routing table should now look something like this:

```

# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
172.16.1.1       0.0.0.0         255.255.255.255 UH      0      0      0 ppp0
166.84.159.0     0.0.0.0         255.255.254.0   U       0      0      0 eth0
127.0.0.0        0.0.0.0         255.0.0.0       U       0      0      0 lo
0.0.0.0          172.16.1.1     0.0.0.0         UG      0      0      0 ppp0

```

where all packets destined outside the local subnet (which is handled by the existing network adapter, eth0) go via Iridium.

You can test the Iridium connection by pinging the gateway:

```

# ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1) from 172.16.8.247 : 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=128 time=1516 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=128 time=1489 ms

```

Note, that ICMP packets are blocked by the gateway, which means that you cannot ping sites other than the gateway, although you should be able to see DNS resolution (i.e., the ping command will translate a domain name into an IP address), which verifies that the connection is working. You can also verify connection to the rest of the Internet by, e.g., attempting to make a ssh connection to a host outside your local subnet.

To terminate the connection, use "ifdown ppp0".

Remote power-on of Iridium phone

The Motorola 9505 phone can not be powered-up simply by turning on the DC power. You need to press the ON button on the keyboard. For remote instrumentation projects this presents a problem. Fortunately, it is possible to reliably power on the phone [using a pair of contacts on its base](#) .

Getting access to the contacts is somewhat difficult, since the DC power adaptors are sealed units. We carefully used a hacksaw to separate the two plastic shells, and inserted a 240 ohm resistor between pin 2 and ground (as is just visible in the photo on the left). [Note: the documentation states that pin 2 should be pulsed low, and returned high; however, it appears to work when held low, and this is the approach we adopted, for simplicity]. Screws were used to return the adaptor to a usable condition, as shown in the second photo.



Automatic dial-back

In order to connect to your remote system at any time, you either need to dial it and establish a PPP connection (a so-called [Mobile-terminated data connection](#)) or simply have it recognise the dial-in attempt, and then dial you back. We took the second approach, and used a small C program to listen to the serial port, waiting for the "RING" response that indicates that the phone is being dialed.

One trick here is that the "RING" response is only given by the phone if the data number is used. If you use the voice number, no response is given.

To-do list

The following needs more attention. Any advice gratefully received.

- [Mobile-terminated data connections](#), i.e., how to dial in to an Iridium phone. Unfortunately, it appears you have to do this from either another Iridium phone, or a modem. It would be nice if it could be done automatically from the Iridium gateway (STOP PRESS: this may be possible in the near future, through a telnet interface).
- IP spoofing, i.e., disconnecting from the Iridium gateway during periods of no communication, and then seamlessly reconnecting at a later time. The Windows software can do this. Maybe it is a matter of requesting the same local IP address when connecting with PPP.
- Compression. It would be useful if the Iridium gateway supported one of the non-proprietary data compression techniques. I haven't tried them all.
- On-demand connection. This is a standard PPP thing, but I haven't tried it yet.
- On-lack-of-demand disconnection. As above.
- Can performance of the PPP link be improved by fiddling with options such as the mru (the Maximum Receive Unit) or mtu (the Maximum Transmit Unit)?

Bibliography

The following relevant documents are on this web site. They have been obtained mostly through google searches, and are placed here in case they disappear from their original sites. It is possible that there are more up-to-date versions available.

- [9505Interface.doc](#) (describes the multi-contact connector on the base of the Motorola 9505 phone)
- [Direct Internet Data Troubleshooting.pdf](#)
- [Application Notes.pdf](#) (general information about Iridium data services)
- [Direct Internet Data Users Guide.pdf](#)
- [AT Command Compliance.pdf](#)
- [Mobile Terminated Data.pdf](#)
- [Dial-Up Data Troubleshooting.pdf](#)
- [AT Command Reference Final.pdf](#) (probably the definitive guide to the Motorola AT command set)
- [Motorola AT Command Set.pdf](#)
- [Dial-Up Data Users Guide.pdf](#)
- [Overview.pdf](#) (an introduction to Iridium data services)

Acknowledgements

A technical person within Iridium was extremely helpful. Telstra were also very helpful.

This page doesn't require Macromedia Flash or Adobe Acrobat. It doesn't have fancy graphics (OK, I couldn't resist the Tux...). Hopefully it gave you the information you wanted, quickly.

Author

[Michael Ashley](#) / m.ashley@unsw.edu.au / last updated 30-Dec-2002