UnixWorld Online: ‘‘PC-Unix Connection’’ Column No. 003

Connect to the Internet with PPP

By Tom Yager

With the advent of the high-speed modem--most recently with the 28.8 kilobit-per-second speed--we entered a new era in telecommunications. Now, over an ordinary phone line, we can link distant systems at about half the speed of a leased line but for as little as 10% of the cost. This raises some interesting possibilities, not the least of which is the chance to put your PC-based UNIX system on the Internet.

The UNIX system provides, among other things, the ability to automatically establish your Internet connection as your system needs it, also known as demand dialing. With some providers this may save you money, but generally you should only use demand-dialed connections with dedicated Internet accounts. The more important benefit of demand dialing is the automatic reestablishment of your Internet link if something should disrupt it.

PPP, the point-to-point protocol, turns a pair of modems into a bridge between networks. Once set up, your system listens for network traffic destined for any remote IP address. If the PPP link is not active, your system will dial your Internet provider’s modem number. After a short negotiation, your PC becomes a full-fledged Internet citizen. As long as the link is live (the modems stay connected), every packet sent to your system from anywhere on the Internet is passed through the modems into your machine. Applications cannot distinguish remote systems accessed through PPP from those on your local network; it’s perfectly transparent.

Your service provider can handle most of the work of putting your PC UNIX system on the Internet. It is your provider’s job to assign you an IP address. If you want to put your entire network on the Internet, you must arrange with your provider to route multiple IP addresses, one for each connected host. This catches lots of people by surprise. You might think just connecting your UNIX host would be enough. It’s not, because your provider’s equipment routes packets for only those systems you’ve paid to connect.

When you arrange for your address assignment, keep in mind that the Internet, unlike the universe, is a finite space. Resources, specifically precious IP addresses, are getting scarce. If you’re connecting an entire LAN to the Internet, think ahead a year or so to the maximum number of hosts you expect to link. The standard in the early days was to carve out a 254-host block of Class C addresses for every site. If you can live with a smaller subnet--my own domain, maxx.net, is assigned a six-host block--you’ll help keep the Internet open to more new users.

If you do pay to route multiple systems, a UNIX host is a perfect gateway. Maxx, my UnixWare system, is a gateway for six machines in my lab. FastLane Communications (my Internet service provider) carved out six addresses--from its assigned pool--which are all mine.

Systems trying to reach yours from the Internet need to be able to translate your host’s name to its assigned IP address. In the next column I’ll describe the process of setting up Domain Name Service

(DNS) at your site. To start, your provider can add your hosts to its name server, so you don't need to run a DNS server of your own.

Basic Setup

While PPP itself is a well-established standard, every UNIX vendor handles setup and configuration differently. They differ enough in their approaches that the best I can do is present a specific case--Novell UnixWare--and leave you to research your system's PPP support on your own.

You can set up PPP without modems. In a pinch, you can use it to create a network connection between two systems using only a serial cable. If you're using a modem, you'll save yourself a lot of time and hassle if you make sure your modem is working properly first. UnixWare requires that you create an entry for your Internet provider's system in your UUCP configuration tables. The modem's serial port must be registered and active under the System Access Facility even though you only dial out.

UnixWare's ''sysadm'' shell will help you set these things up. You'll also find some X Window System point-and-click tools. Like me, you may prefer to hack up some of the files, particularly those in `/etc/uucp`, by hand. At the very least, check the automated tools' changes. I've had UnixWare's tools create double entries in the port monitor tables, creating havoc with my modem pool. All the more reason to understand your system at as deep a level as you can.

Your Internet provider's equipment has a text protocol it uses to greet, collect a username and password, and then report the connection active. You need to document this protocol and turn it into a chat script. You should only have to carry the script as far as the password; PPP's own negotiating process takes over from there. If you've written UUCP chat scripts before, then you understand how this works. Otherwise, think of it simply as a ''you say, I say, you say...'' sequence. Consider the following Internet provider chat script lifted straight from Maxx's `/etc/uucp/Systems` file:

```
fastlane Any V34PPP 110 5551212 ogin:-\r-ogin: fred ssword: bonehead
```

The provider's name, the acceptable time to make an outgoing call and the modem class (from `/etc/uucp/Devices`) are the first three entries. The next entry (110) is the serial port speed. The Equinox SuperSerial card I use to drive Maxx's modems aliases useless baud rates to higher rates (57,600 and up) not supported by System V's limited speed table. The next entry is the phone number, and then the chat script takes over.

Note that a cracker could use data in the `Systems` file to make your life lots more interesting. Make sure all the configuration files in `/etc/uucp` are heavily protected; only the ''uucp'' user and group should have *any* access to these files.

After you've set up `Systems` and other UUCP configuration files you should be able to use `cu fastlane` (of course, use a system name descriptive of your provider) to make a connection. This isn't a PPP link yet, but if your system dials out and connects to your provider's host you've got half the battle won. Note that `cu` won't run the chat script, either. You'll have to wait a few steps before you can check the chat.

UnixWare provides a handy script named `/usr/sbin/pppconf` that asks all the right questions. You need to have a working UUCP entry for your provider before you run the script. For outgoing

connections, use the `pppconf` script to add a PPP host. You'll be prompted for IP addresses for both sides of the connection, local and remote. Your provider will give you these.

As is common with other commercial SVR4 Unix offerings, UnixWare cannot deal with accounts that have floating IP addresses. The local and remote addresses must be known in advance. Although, PPP has the ability to negotiate these addresses, UnixWare's PPP implementation does not use this feature. Linux is one example of an operating system which *can* deal with floating IP addresses, but then Linux doesn't yet dial on demand (when I checked last).

There are a number of parameters that govern how PPP behaves. For instance, it has the ability to compress data for greater throughput in a given time. You can also tune packet size and other parameters. Those parameters that matter to your provider should be part of the information they give you. If you don't get this information, run with the defaults and ask your provider for details if performance seems poor.

Make the Link

If you've done all this setup correctly--and you've lived a clean life up to now--you should be ready to go on the air. You have to reboot your system first because UnixWare initializes all your PPP parameters at boot time. (That's why it can't support dynamic IP addresses.)

UnixWare automatically sets up your PPP link as your default gateway. What this does is forward every packet destined outside your local network to the Internet via PPP. If your PPP link will rise and fall under normal operation, you may need to create a script that undefines this gateway. If the gateway is defined but the PPP link is down, you may find your network bogs down. I created a couple of simple scripts that let me raise and lower the link at will:

```
#! /bin/sh
# @(#) pppup: set up PPP connection
ifconfig ppp0 up
route add default pppgateway 1

#! /bin/sh
# @(#) pppdown: tear down PPP connection
route delete default pppgateway
ifconfig ppp0 down
```

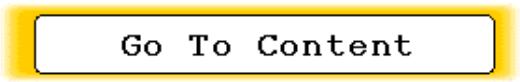
The `pppgateway` is an entry in `/etc/hosts` that identifies the IP address of the remote side of the link (your Internet service provider).

Just raising the link as shown above (UnixWare will raise it at boot time unless you remove the initialization code) isn't enough to make your system dial out. That requires outbound traffic. A simple `ping` destined for some remote host will do. If you want to keep your link active you should add a `cron`-table entry that performs a one-packet `ping` to your provider's host. The equipment your provider uses, and perhaps your own modem, may time out if there's no traffic for a period of time.

If you're not exporting any services (like FTP or HTTP) to the Internet, it may not matter that your link isn't always up. If you want electronic mail delivered directly to your host you'll also probably want to keep your link hot. Just keep in mind that demand dialing works only for *outbound* packets. Your system can't know if there is traffic intended for it unless the link is live.

Next month I'll cover the process of setting up a Domain Name Service server on your system. As always, thanks for reading.

Copyright © 1995 The McGraw-Hill Companies, Inc. All Rights Reserved.
Edited by Becca Thomas / Online Editor / UnixWorld Online / beccat@wcmh.com



Last Modified: Thursday, 23-Nov-95 07:49:10 PST