DSL HOWTO for Linux

Hal Burgiss

hal@foobox.net

Original Author: David Fannin

dfannin@sushisoft.com

Edited by

Greg LeBlanc

v1.3, 2001−06−25

Revision History

Revision v1.3	2001−06−25	Revised by: hb
Updates to various sections.		
Revision v1.2	2001−03−28	Revised by: hb
Assorted changes and additions.		
Revision v1.1	2000−11−14	Revised by: hb
Many miscellaneous minor corrections and updates.		
Revision v0.99	2000−09−05	Revised by: hb
Various updates, additions and new sections.		
Revision v0.92	1999−04−10	Revised by: df
First release (ADSL mini HOWTO).		

This document examines the DSL family of high speed Internet services now being deployed in various markets worldwide. Information is included on the technology behind DSL as well as subscribing, installing, configuring, and troubleshooting, with an emphasis on how this impacts Linux users.

Table of Contents

Table of Contents

Table of Contents

1. Introduction

DSL, or Digital Subscriber Loop, is a high−speed Internet access technology that uses a standard copper telephone line (a.k.a. "loop" in telco parlance). DSL provides a direct, dedicated connection to an ISP via the existing telco network. DSL is designed to run on up to 80% of the telephone lines available in the United States. By using line−adaptive modulation, DSL is capable of providing data speeds of 8 Mbps, or more.

DSL services are now being aggressively marketed for home and small business use. DSL is typically priced below ISDN, and well below T1 service, yet can provide potentially even greater speeds than T1 without the cost, complexity, and availability issues of T1. Since DSL is a dedicated, often "always on" service, it avoids the delays and use charges that are common with ISDN. Making this quite a nice technology for the bandwidth starved masses.

While all this sounds exciting, DSL does have some drawbacks. The quality of the DSL signal, and thus the connection, depends on distance (the length of the copper "loop") and various other factors. Also, there is no such thing as standard "DSL". There are various flavors of DSL, and many, many ways DSL providers are implementing their networks. In typical fashion, Linux users are often left to fend for themselves, since the DSL providers are often taking the easy way out, and catering only to "mainstream" Operating Systems.

The topics included in this HOWTO include qualification and pre−installation, installation, configuration, troubleshooting and securing a DSL connection. As well as other related topics. There are also appendices including a comprehensive DSL Overview, Frequently Asked Questions, a listing of related links, and a glossary.

Due to the fast pace of change in the telco and DSL industries, please make sure you have the latest version of this document. The current official version can always be found at http://www.linuxdoc.org/HOWTO/DSL−HOWTO/. Pre−release versions can be found at http://feenix.burgiss.net/ldp/adsl/.

1.1. Document Structure and Reading Guidelines

This document attempts to give a comprehensive discussion of DSL. All aspects are hopefully addressed to one degree or another with what can be a complex topic since it deals with networking, hardware, new fangled technologies, and various approaches taken by various vendors. The core components of this document are:

- The Installation section covers installation of DSL hardware and related components, including wiring considerations, splitter or microfilter installation, modem and Network card installation.
- The Configuring Linux section covers mostly client and software aspects of getting the connection up and running. The Network card configuration is actually covered mostly in the above Installation section.
- The Securing Your Connection section covers Security implications that are even more important with a full−time connection. Linux users seem especially targeted by crackers, because quite frankly, some don't understand how important security is, or don't understand the finer points of this.
- The Tuning and Troubleshooting section covers post−installation topics like how well is our connection performing, and how to track down any show−stoppers or intermittent problems.
- There is also a lengthy Appendix that covers various topics relating to Linux and DSL. None of these are directly related to simply getting that connection up and running, but may be of interest

nonetheless.

To simplify the navigation of this document, below is a suggested reading guideline. Everyone should read the Introduction. Please pay special attention to the Conventions and Terminology section, as some of this terminology may be used somewhat differently in other contexts. Also, there is a Glossary if you get lost in the world of TA (telco acronyms) ;−).

- If you don't know anything about DSL, you should probably read the entire document. You may want to start with the DSL Overview section in the Appendix, and then the FAQ. The DSL Overview explains how the various pieces of the puzzle fit together. DSL network implementations are more complex than traditional dialup networks.
- If you have already done some homework, but have not ordered service from anyone yet, read the Choosing Providers section, and the Linux Friendly ISPs sections. Also, you might get a head start by reading the Configuring Linux section so you know what lies ahead.
- If you have ordered service already, and are awaiting delivery, you can skip the sections on choosing a Provider. If you will be doing a self−install, you should read the pertinent parts of the Installation section, the Configuring Linux section, and the Securing Your Connection section.
- If the installation is complete, and you can't get a working connection, skip right to the Troubleshooting Section. If you are not clear on what protocols are required, or what software you need to have installed, also read the Configuring Linux section. If not sure what terms like "sync" mean in this context, then be sure to read the DSL Overview section first so you know how it all fits together.
- If trying to decide between cable and DSL, read the Cable vs DSL section, and possibly the DSL Overview section.
- If you have never had a full−time Internet connection, or are not absolutely sure you fully understand how to secure you connection, be sure to read The Securing Your Connection section. If you don't understand some aspect of this, re−read it, or start looking for other references.
- There is a comprehensive Links section that has references to some topics not touched on in the main body of the Document itself.

1.2. What's New

Version 1.3: Updates to the SpeedTouch USB HOWTO in the appendix. Minor update to PPPoE section, and two new Linux Friendly ISPs. A feeble attempt to make the document a little less U.S.−centric. Various minor updates.

Version 1.2 adds PPTP configuration section for Alcatel ethernet modems. Also, added are two additional sections in the "Tuning" section for the TCP Receive window, and ADSL/DMT interleaving. And the big news is the release of open source drivers for the Alcatel USB modem as of March 2001. There is an Alcatel SpeedTouch USB mini HOWTO in the appendix by Chris Jones. A number of miscellaneous updates as well.

Version 1.1 included quite a few minor corrections, updates, and additions. Not much that is substantially new. There are finally two Linux compatible DSL PCI modems from Xpeed. The drivers are now in the kernel 2.2.18 source.

Version .99 addresses some of the many changes that have occurred since the original ADSL mini HOWTO was published. Originally, ADSL was the primary DSL technology being deployed, but more and more some of the other DSL flavors are entering the picture −− IDSL, SDSL, G.Lite, and RADSL. Thus the renaming from "ADSL mini HOWTO" to the "DSL HOWTO". There have been many other changes in DSL technology as well. PPPoE/A encapsulation has become more and more common as many ISPs are jumping on this bandwagon.

1.3. Copyright

1.4. Credits

Thanks to all those that contributed information to this HOWTO. I have anti−spammed their email addresses for their safety (and mine!). Remove the X's from their names.

- *B Ediger* (Xbediger@csn.net) Great Description of loop impairment.
- *C Wiesner* (Xcraig@wkmn.com) List of many ADSL URLs.
- *J Leeuw* (Xjacco2@dds.nl) Many tips on ADSL, especially in Europe
- *N Silberstein* (Xnick@tpdinc.com) Info on Netrunner and his experience with US Worst.
- Many and various posters from comp.dcom.xdsl and bellsouth.net.support.adsl, too numerous to mention individually. (HB)
- *Juha Saarinen* for suggestions and explanations on the TCP Receive Window, and related tuning topics.
- *Chris Jones* <chris@black−sun.co.uk> for his Alcatel SpeedTouch USB mini HOWTO which has been incorporated into the Appendix.

1.5. Disclaimer

The authors accept no liability for the contents of this document. Use the concepts, examples and other content at your own risk. As this is a new edition, there may be errors and inaccuracies. Hopefully these are few and far between. The author(s) do not accept any responsibility for incorrect or misleading information, and would certainly appreciate any corrections. Also, this type of technology dates itself very quickly. What may be true today, is not guaranteed to be true tomorrow.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

References to any particular product, brand, service or company should not be construed as an endorsement or recommendation.

1.6. Feedback

Any and all comments on this document are most welcomed. Please make sure you have the most current version before submitting corrections! These can be sent to <hal@foobox.net>

1.7. Conventions, Usage and Terminology

For the sake of simplicity and sanity, let's clarify some of the terminology that we will be using in this document, so that we are all on the same page. While many of the definitions below are not always 100% technically correct, they are close enough for our purposes here. In fast moving technologies like DSL, there are so many "ifs, ands, and buts" that it is difficult to say anything with any degree of certainty and have it stick. And there are exceptions to almost every rule. And sometimes exceptions to the exceptions. We will be dealing with generalities to a large degree here, please keep that in mind.

- "DSL" will be used to refer to the entire family of DSL technologies now available −− ADSL, SDSL, IDSL, RADSL, etc. ADSL still seems to be the most prevalent at this time, but the others are being deployed more and more. Where it is important to differentiate one type of DSL from another, the full proper name will be used: e.g. RADSL. xDSL is also commonly used to refer to the various DSL technologies as a group, but we will be using just "DSL" here.
- The term "telco" here refers to any potential DSL provider. This includes the ILECs (Incumbent Local Exchange Carriers), a.k.a. the old guard phone companies or state run phone companies, and where the monopolies now have competition, the CLECs (Competitive Local Exchange Carriers), or independent providers such as Covad and Rhythms in the U.S.
- "CO" is the telco acronym for "Central Office". Traditionally this is a building where one end of your phone line physically terminates. The other end terminates at your home, office, or wherever. It will be used here to refer to the telco end termination point, regardless of whether it is a traditional Central Office building or another, smaller, remote structure or device.
- "Loop" is telco speak for "phone line". Essentially, you should think of your loop as one dedicated pair of copper wires that run uninterrupted from your residence or office directly to the CO. This is perhaps an oversimplification, but will serve our purposes. DSL availability, and signal quality, is tied directly to the characteristics of your physical line −− or "loop" as they say.
- "POTS" is the acronym for Plain Old Telephone Service. In other words, traditional, non−digital devices like phones, faxes and answering machines.
- "NID", or Network Interface Device, is the small telco housing that is often typically attached to the outside wall of your house, and is the service entrance for telco services, though may be placed elsewhere depending on the phone company. This may variously also be referred to as "ONI", "SNI", "NIU", "TNI" or other creative telco acronyms. It represents the "demarcation" point that divides the customer's realm of responsibility from the telco's. Commercial structures, and multi−family housing will likely have something more sophisticated, and probably located inside somewhere.
- "DSLAM" is the sophisticated hardware device in the telco's CO where your phone line physically terminates, and thus makes DSL happen. Increasingly, telcos are making use of smaller devices like the "mini−RAM" in remote locations. We'll use "DSLAM" here as a catchall for any device that enables DSL service from a telco. These are now being manufactured by a number of companies.
- "Modem" will be used to refer to the end user device that enables a DSL connection. Your modem is

connected to the telco's DSLAM in the CO via your loop. When they are "talking" DSL to each other, they are in "sync". Without "sync", no connection to your ISP is possible.

"Modem" is indeed the correct terminology since there is MOdulation and DEModulation of the signal, even though it isn't strictly speaking an analog device like many of us have had before. These modems typically have other features too. Some ISPs and manufacturers may be marketing simply "routers", "bridges", or even "brouters" for this purpose. These are essentially DSL modems with enhancements. A compatible "modem" of some kind is the minimum hardware requirement at the customer's end of the connection. The most commonly supplied modem is actually a combination bridge and modem.

Unless stated otherwise, we will also be assuming the "modem" has an ethernet interface, and will connect to a standard ethernet Network Card (NIC). This is far and away the most prevalent configuration, at least until more Linux drivers are available for PCI and USB modems.

It is worth noting that "routers" as supplied by DSL providers are typically modem/router combination devices. In our context, "router" will refer to these devices as such. There are also SOHO broadband routers available that are only dedicated routers and lack the modem functionality.

- Previous versions of this document referred to the modem as an "ANT" (ADSL Network Termination). While this may be technically correct terminology, it is not used by ISPs, manufacturers, telcos, or most users to any extent. The "modem" will be just called a modem, regardless of whatever other features it may have (i.e. router, bridge, etc.).
- PPPoX will be used to refer to PPPoE (PPP over Ethernet) and PPPoA (PPP over ATM) collectively. These protocols are being used by many DSL providers now.
- The information provided in this document is based on the current state of DSL in the U.S. I would assume there are enough similarities with DSL services outside of the US that this document would still have some merit for everyone. Correct me if I am wrong by emailing <hal@foobox.net>.
- A "#" will be used to denote a command that typically is run by the root user. Otherwise, a "$" will be used as the prompt for non−root users.

2. Installation

Before actually ordering service, there are several things you may want to explore. Please note, that there are many ways any given telco might decide to handle qualification and installation procedures. Much of what is described in this section, is how it is commonly done in the U.S.

2.1. Pre−Installation

In many parts of the world, there is no choice on who you get DSL from: your friendly local telco, of course! They own the copper wires, and thus they hold all the cards.

However, in the U.S. de−regulation has opened this up somewhat. Beyond the obvious consideration of price, there are many reasons to investigate which alternate providers may be offering DSL services in your area. The large Telephone companies are everywhere, and may advertise the most. But increasingly smaller ISPs and independents are getting into the act. This is creating diversity in the DSL marketplace. A good thing of course, but possibly creating a little confusion too. Conversely, in areas where there is only one choice, then we have no choice but to accept whatever service is being offered.

If your telco has a monopoly on phone service and DSL, you may skip the rest of this section. And probably the next few sections. They will probably control the installation and qualification processes, and you just wait for them to get finished.

Not all DSL services are alike. Just because two local companies are offering "ADSL", does not mean that necessarily there is much in common at all. In fact, there are potentially a number of factors that make one ADSL provider's service significantly different from another's. Some things to consider:

- Speed vs Price.
- What hardware is provided, i.e. modem or router. It is best if this is external ethernet in either case.
- The ISP's Network architecture. PPPoX? Static IP? Servers allowed?
- Is it an "always on" service, at least theoretically? Are there supplemental usage fees, or idle timeouts?
- Linux friendly, Linux hostile, or Linux agnostic?
- Quality of service. How is news, mail, etc.?

For a more lengthy discussion on some of these considerations and related issues, see the DSL Overview appendix for more on modems, qualifying for service, and choosing a provider.

Once you have chosen a provider, and ordered service, the next step is for the telco to "qualify" your loop. This essentially means testing your line to make sure it can handle the DSL signal, and possibly what level of service may be available to you. This may take some time, especially if the telco encounters problems with the loop. If no problems are found during this phase, then possibly there will be a two to three week wait for the installation. YMMV.

After the telco has qualified the loop and readied their end of the connection, the next step is installation of the necessary components at the customer's end of the connection: wiring modifications, splitter or filters, and, of course the modem and any necessary software.

2.2. Installation Options −− Self Install or Not

You may or may not have a choice on how the installation is done, or who does it. This is totally at the discretion of the provider. In much of the world, this is done by the telco, and there is little flexibility. Many providers in the U.S. offer a "self install" option where you do all the work. In this scenario, the provider will send a kit in order to save them from sending a tech, and thus reducing cost. Typically, self install kits will include microfilters for the POTS (Plain Old Telephone Service) phone jacks, the modem (and maybe a NIC), and a CDROM with drivers, etc. on it. In some cases, a splitter may be included instead of microfilters. In any case, some type of filtering is necessary on the non−DSL lines. If not the noise generated by the DSL signal may interfere with POTS devices.

The other possibility is for the provider to do the installation. Again, this may be your only option. Obviously, the cost is higher here, but it may have the advantage of having a trained tech do any wiring. There is also a better chance of getting a "splittered" installation with this option (a good thing!). Another benefit is that if something is wrong with the line, or the telco has not provisioned the line properly, an on−site tech may be able to help sort out certain kinds of problems quickly.

The self−install kit should come with full instructions, regardless of whether the installation will be splittered or filtered. So we won't go into much detail on this aspect.

2.3. Wiring/Installation Options

There are various wiring schemes depending on how your service is being provided, who is providing it, and which DSL service is being provided. If your telco is performing the installation, you may skip this section.

- *Dedicated Line*. Some DSLs require a dedicated, or "dry", wire pair, e.g. IDSL. This means a separate, physical line without dial−tone for DSL and Internet connectivity. Also, DSL services from CLECs (independent telcos like Covad or Rhythms), may use a dedicated line, depending on their line sharing agreement with the local incumbent carrier. (Instead the CLEC will actually lease a loop from the ILEC.) On your end, this simply means using one of the unused wire pairs in the telco wire bundle, and connecting it to the DSL jack.
- *Shared Line with Splitter*. For DSLs like ADSL, that are provided over the same line as regular voice service (POTS), the signal must be filtered somehow so that voice services are not adversely effected. Installing a splitter splits the line into two pairs, and filters the DSL signal from one of them. This results in a inside wiring scheme where DSL goes to only one jack, and then POTS service to all other jacks. This is considered by many to be a better type of installation than "splitterless", i.e. with microfilters instead. See below.

  Splitters are available from various manufacturers and come in various shapes and sizes. Some are small enough to fit in the NID itself (sometimes called SNI, this is the telco phone box on the outside of your house), while others have a housing as large as the NID itself. Typically this is mounted near the NID, on the customer's side of the demarcation point.
- *Shared Line with Filters*. Again, for some DSLs that piggyback on the POTS line, the signal must be filtered or split at some point. This is not necessary for g.lite or RADSL however. The other way of doing this is by placing RJ11 "microfilters" in each phone jack −− *except where the DSL modem will be*. These filters are relatively small, plug−in devices and remove the higher frequencies associated with DSL. This is obviously much easier since no tools or wiring is required. This is often what is

included in self−install kits, and is often referred to as a "splitterless" installation. This is a very common approach in the U.S.

Similar microfilters are sometimes used by some telcos to reduce the excessive "whine" on the line that is produced by some modems. This is a little different approach as the filter is put on the same jack as the modem.

- *Shared Line, Splitterless and Filterless*. Some newer DSLs, like G.Lite, have no adverse effect on regular POTS devices and thus require no filters or splitters. This would seem to be the wave of the future. Just plug and play. Though still not very common.

Figure 1: DSL Block Diagram, POTS with Splitter (NID not shown)

```
      <--------Home/Office-----><---Loop---><--Central Office-->

POTS  X-------+
phone,        |
fax,          |
etc,          |
              |                                 CO
              |                               -------
              |                               |     |
              |                               |     |
              |            -----              |     |
POTS  X-------+----Voice--=| S |              |  D  |
                           | P |              |  S  |=- Voice Switch
                           | L |    2 wire    |  L  |
                           | I |=------------=|  A  |
                           | T |  Local Loop  |  M  |=- ISP --> INET
          ---------        | T |              |     |
Linux X--=| Modem |=-Data-=| E |              |     |
          ---------        | R |              |     |
                           -----              |     |
                                              -------
```

Figure 2: DSL Splitterless (a.k.a. filtered) Block Diagram

```
      <--------Home/Office-------><----Loop---><--Central Office-->


POTS  X--Voice---[RJ11]------+
phone,          (filter)     |
fax,                         D                      CO
etc,                         a                    -------
                             t                    |     |
                             a                    |     |
POTS  X--Voice---[RJ11]----- &                    |  D  |
                (filter)     V  -----             |  S  |=- Voice Switch
                             o  | N |   2 wire    |  L  |
                             i-=| I |=-----------=|  A  |
                             c  | D |  Local Loop |  M  |=- ISP --> INET
                             e  -----             |     |
          -----------        |                    |     |
Linux X--=|  Modem  |=-------|                    |     |
          -----------                             -------
```

2.4. Self Install − Wiring

If you are not doing a self−install, then you may skip this section and move to Configuring Linux. If you are doing a self−install with microfilters, skip to the mircofilter section. The following procedures are meant to illustrate the wiring process. Please note that your procedures may be different at your location. Make sure you follow any warnings or safety instructions provided, that you RTFM, and that you are familiar with telco wiring procedures.

The first step will be to wire up the connections from your provider. Identify the line on which service will be installed, and the locations of your splitter and DSL jack(s). (For perhaps a better wiring scheme, see the Homerun section immediately below.)

Be aware that typical telco wire has more than one pair per bundle. Often, two pairs, but sometimes more. If you have but one phone line, the other pair(s) are unused. This makes them available for use with wiring for DSL. Wire pairs are color coded for easy identification. SDSL and IDSL require a dedicated, or "dry", pair. If an unused pair is available, then no real re−wiring is required. It is just a matter of re−wiring an existing jack for the correct pair of wires, and attaching the modem.

2.4.1. The Homerun

> " *I would not use microfilters if I lived across the street from my CO. A splitter is the only way to go.* "
>
> −−A retired BellSouth ADSL installer

The optimum method of wiring for the DSL modem is sometimes called a "homerun". It is called this because it is one, straight shot from the splitter to the modem's DSL jack. What this does is bypass the existing inside wiring altogether, and any problems that might be lurking there −− like a corroded connection somewhere on a POTS jack. Inside wiring deficiencies can cause a degradation of the DSL signal.

This also allows you to route the cable to avoid any potential RFI (Radio Frequency Interference) sources. RFI anywhere in the circuit can be a DSL killer. Routing the cable away from items that may have electric motors, transformers, power supplies, high intensity lighting fixtures, dimmer switches and such, is a smart way to go. And you are also less likely to have a failing microfilter cause problems −− one potential point of failure instead of several. You can also use a better grade of cable such as CAT 5.

If your existing installation is "splitterless" (i.e. using microfilters) now, converting to a homerun will entail purchasing a splitter. And, of course, will also mean some new wiring will need to be run. Microfilters also add to the effective loop length −− as much as 700 ft per filter in some cases! So if you have several microfilters installed, and your sync rate or distance is marginal, eliminating these filters may result in a significant improvement.

A poor man's splitter can be rigged by using a microfilter inside the NID. This is not "by the book", but seems to work just fine for many.

2.5. Wire the Splitter

If you have the splitterless design (i.e. using "microfilters") or a dedicated line, you may skip this part.

The splitter will typically consist of two parts, the splitter and a small outdoor housing. Mount the splitter and accompanying housing per the telco's instructions at the Network Interface Device (NID) point (also sometimes called the SNI or ONI), usually the side of your house where the phone line is located. Put it on your side of the NID. The phone company may need to access the splitter for maintenance, so its advisable to locate it on the outside where they can get at it, but outside is not absolutely necessary.

The wire bundle should have at least two separate wire pairs. The splitter takes one pair, and separates the signal onto two pairs. One pair in the bundle will then go to all POTS jacks, and the other to the modem's DSL wall jack. So connect the incoming telco line to the LINE side of the splitter. Then wire the inside pair for your telephone to the VOICE, and your inside wire pair for the modem to DATA.

Checkstep At this point, you should be able to pull dial tone off the voice side of the splitter. If this doesn't work, then you've wired it wrong. You can also plug the modem into the test jack in the NID box (most should have this). Plug in the modem's power cord, and if the line is provisioned correctly, you should "sync" in less than a minute. This test only requires the modem. (Internal and USB modems will require a driver to be loaded before syncing. This would mean having the computer there too.)

2.6. Wire the DSL Jack

Wire the DSL wall jack (RJ11) at your computer location, which should already be connected to the DATA side of the splitter. The specifics differ for each situation, but basically you will have a wire pair that you will connect to the DSL jack. Make sure you *read the directions*, as the DSL−RJ11 wiring may be different for phones and DSL jacks. *AND* −− different modems may expect the signal on different pairs −− most on the inside pair, but some on the outside pair.

Figure 3: RJ11 Wiring options

```
      ||
      ||
      ||
     /  \
    |RJ11|
    |    |
----
||||

 ^^  <-- Inside     Most modems on inside pair
^  ^ <-- Outside    Some on outside, e.g. Alcatel 1000, SpeedTouch Home
```

2.7. Installing Microfilters

Pretty much a no−brainer here. If you are doing a "splitterless", self−install installation, then install the provided microfilters in all phone jacks *except* the one where the DSL modem will be connected. Don't forget devices like fax machines and analog modems. The filters filter out the higher DSL frequencies and will keep the DSL noise from interfering with POTS equipment.

Warning! If you have an alarm system, it is recommended not to use microfilters. Alarm systems can present various problems, depending on the type of alarm and how it is installed. The recommended installation in this case is with a splitter.

2.8. Installing an Ethernet Modem

To install, connect the modem's (or router's) power cord, and connect the phone line between the DSL wall jack and the modem. This cable should be provided. If not, a regular phone cord will suffice. With the ethernet interfaced modems, you may also connect the ethernet cable between the NIC and the modem (but not really necessary at this point just to verify an ethernet modem is working).

Checkstep At this point, verify that the modem syncs with the telco's DSLAM signal. Most modems have a green LED that lights up when the signal is good, and red or orange if not in sync. The modem's manual will have more details on the LEDs. If it doesn't sync, then check your wiring, or make sure that the DSL signal is being sent. Do this by calling your telco and verifying they have activated the service. Or by testing the modem at the test jack on the NID (see above). Note that having dial tone on the line does NOT confirm the presence of the DSL data signal. And vice versa −− perfectly possible to have dial tone and no DSL, or DSL and no dial tone. There should also be no static or noise on the voice line when everything is installed and functioning properly.

2.8.1. Installing the Ethernet Network Card (NIC)

Ethernet modems will, of course, require an ethernet network card. If you haven't already done so, install the NIC in your Linux machine, configure the kernel, or load modules, etc., etc. This is sometimes the biggest stumbling block −− getting the NIC recognized and working. See the various Linux references for doing this, such as the Ethernet HOWTO for more information. Also, see the Troubleshooting Section below. This is certainly something you could conceivably do ahead of time if you already have the NIC.

Be sure the RJ45 cable between the NIC and the modem is now connected. You can "hot plug" this cable, meaning there is no need to power down to do this.

We can do a few quick tests now to see if the NIC seems to be functioning properly. First we'll attempt to bring up the interface. Then we'll see how well it is responding by **pinging** it. And lastly use **ifconfig** to check for errors:

```
# ifconfig eth0 10.0.0.1 up


$ ping −c 50 10.0.0.1
PING 10.0.0.1 (10.0.0.1) from 10.0.0.1: 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=0 ttl=255 time=0.2 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=255 time=0.1 ms
<snip>

− 10.0.0.1 ping statistics −
50 packets transmitted, 50 packets received, 0% packet loss
round−trip min/avg/max = 0.1/0.1/0.2 ms


$ ifconfig eth0
eth0    Link encap:Ethernet  HWaddr 00:50:04:C2:09:AC
        inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:428 errors:0 dropped:0 overruns:0 frame:0
        TX packets:421 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        Interrupt:10 Base address:0xc800
```

If "eth0" comes up without errors, and you can **ping** it without errors, and **ifconfig** shows no errors, we most likely have all our hardware in working order now, and are ready to start configuring Linux. If not, see the Troubleshooting section below.

Gotcha: A few modems may already be wired as a 10baseT crossover, and require a direct Category 5 cable for a direct connection to a NIC, rather than a crossover cable. I lost around 12 hours figuring this one out, so don't make the same mistake − make sure you RTFM first.

2.9. Installing a USB Modem

The physical installation of a USB modem is similar to an ethernet modem. There is no ethernet card necessary obviously. So connect the phone line between the DSL wall jack and the modem's DSL port, and attach the USB cable to the computer's USB port.

USB modems will require vendor and model specific drivers in order to sync and function properly. Assuming you are using the Alcatel SpeedTouch USB, then there are drivers available from Alcatel's driver page: http://www.alcatel.com/consumer/dsl/supuser.htm#driver. This driver is a kernel module, and is available as both binary and raw source. This driver requires at least a 2.4.1 kernel. In addition, you will need to install additional software and patch the kernel for other required features. There are brief instructions included with the driver. Much of the code is still experimental, and reportedly is not completely stable at this time. This is not novice level stuff!

This driver also supports both PPPoE and PPPoA, though the steps for getting either to work are quite different. See the Appendix for PPPoA step−by−step instructions.

3. Configuring Linux

After you have connected the modem and it's getting sync, then you're ready to configure Linux and verify your connection to your ISP. Although I will refer to a Linux System, you could conceivably connect any type of 10baseT device to the modem. This includes a router, hub, switch, PC, or any other system that you wish to use. We'll just cover the Linux aspects here.

Warning

Before you connect to your ISP, make sure you understand all security issues of having a direct connection to the Internet via DSL. Depending on your ISP, most outside users can access your system, and you should setup any firewalls, deactivate ports/services, and setup any passwords prior to connecting your machine to the world. See the Security section below, and the links section for more on this *very important* topic. Do not make this an afterthought! Be ready.

3.1. Bridged vs PPPoX Networks

Before we get too far into the final stages of installing and configuring our system, let's look at how various DSL ISPs set up their networks. It will be very important for you to know how your ISP does this, as there is more than one possibility and the steps involved are quite different for each. This may not be the kind of thing the ISP is advertising, and since you are not using Windows, you may not have access to the setup disk that the ISP provides. If you're not sure, ask the ISP's tech support staff, or other users.

To muddy the waters even more, some ISPs may be offering more than one kind of service (over and above the various bit rate plans). Example: Verizon (formerly Bell Atlantic) originally offered static IPs with a Bridged connection. Now all new installs use PPPoE with dynamic IPs. For installation and configuration purposes, this is very different.

The two most common DSL network implementations are Bridged/DHCP and PPPoX. Both have mechanisms for obtaining an IP address and other related networking configuration details so we shouldn't have to worry about this. But there are indeed other, less common, means of connecting. Our job will be finding the right client, and doing what we have to, to get it up and running. The most common ones are discussed below.

Important! You need to know beforehand how your ISP is setup for connecting to his network. To re−iterate, the two main possibilities are Bridged/DHCP and PPPoE. These are mutually exclusive implementations. And there are indeed other possibilities as well. So you will need to know exactly what this is beforehand. And it must be the right one or you will waste a lot of time and effort. You cannot choose which one either. It is a matter of how the ISP is doing his network. Note that PPPoE can run over Bridged networks, so just knowing whether you are Bridged or not, is not necessarily good enough. If your provider is giving you a router, there is a good chance that the router's firmware will handle all of this for you.

If you are subscribing with one of the Baby Bells in the U.S., you can probably count on that being PPPoE, and thus you will need a PPPoE client.

There are a few provider specific FAQs and HOWTOs in the Links section below.

3.1.1. Bridged/DHCP

In the good old days of a year or two ago, purely "Bridged" connections were the norm. PPPoE had not been invented yet. This type of network puts you on a local subnet just like a big LAN. You are exposed to much of the local subnet traffic, especially ARP and broadcast traffic. The typical means of authenticating in this set up, is via DHCP.

DHCP is a standard, established networking protocol for obtaining an IP address and other important network parameters (e.g. nameservers). This is a standard, well documented networking scheme and is very easy to set up from the end user's perspective. It is also a very stable connection. You can actually unplug the modem for say 10 minutes, plug it back in, let it re−sync, and the connection is still there −− same IP and everything.

3.1.2. PPPoX

The main alternative now is PPPoX, meaning either PPPoE (PPP over Ethernet) or PPPoA (PPP over ATM). Both of these related protocols are currently being deployed, but at the moment, PPPoE seems to be the more common of the two. PPPoX is a relative newcomer, and, as the name implies, is a variation of Point−to−Point Protocol that has been adapted specifically for DSL networks.

There are several PPPoE clients for Linux (see below). PPPoX simulates a dialup type environment. The user is authenticated by user id and password which is passed to a RADIUS server, just like good ol' dialup PPP. A routable IP address, and other related information, is returned to the client. Of course, no actual dialing takes place. The mechanics of how this is handled, will vary from client to client, so best to RTFM closely. Typically you will set up configuration files like `pap−secrets`, etc.

It is worth noting that PPPoE will also work on non−ethernet devices like USB, provided the correct drivers are installed.

From the ISPs perspective, PPP is much easier to maintain and troubleshoot. From the end user's perspective, it is often more work to set up, often uses more CPU, and the connection is maybe not as stable. So anyway, this seems to be the coming trend. Many of the large telcos around the world, especially the RBOCs (Baby Bells) in the U.S., have committed to PPPoX already. Setting up a PPPoX connection is completely different from setting up a bridged/DHCP connection.

3.1.3. ATM

Since the traffic on the wire from the DSLAM to the modem is typically ATM, a raw ATM connection would seem to make sense. While possible, this is rare, if it exists at all in the U.S, and would require a modem in addition to a PCI ATM card, such as the Efficient Networks 3010. There is an ATM project for Linux, that is being incorporated into the 2.4 kernel. (See the Links section for more information.)

This may be a viable solution at some point, but it is just not "there" yet.

3.2. Configuring the WAN Interface

The most common configuration is a DSL modem in "bridging" mode. Both PPPoX and DHCP can use this setup. In this scenario, the WAN interface typically means your NIC. This is where your system meets the outside world. (If you have a router see below for router specific instructions.) So essentially we will be configuring the NIC, typically "eth0" since it is an ethernet interface.

With PPPoX, once the connection comes up, there will be a "ppp0", or similar, interface, just like dialup. This will become the WAN interface once the connection to the PPP server is up, but for configuration purposes we will we be concerned with "eth0" initially.

There are various ways an ISP may set up your IP connection:

- Static IP.
- Dynamic IP on Bridged Network via DHCP.
- Dynamic IP via PPPoX.
- Static IP via PPPoX.

Let's look at these individually.

3.2.1. Static IP Configuration

A "static" IP address is an IP that is guaranteed not to change. This is the preferred way to go for those wanting to host a domain or run some type of public server, but is not available from all ISPs. Note that while there are some noteworthy benefits to having a static IP, the disadvantage is that is more difficult to remain "invisible". It is harder to hide from those with malicious intentions. Skip this section if you do not have a static IP, or if you have a router, and the router will be assigned the static IP.

Configure the IP address, subnet mask, default gateway, and DNS server information as provided by the ISP. Each Linux Distribution (Redhat, Debian, Slackware, SuSE, etc.) has a different way of doing this, so check on your distro's docs on this. Each may have their own tools for this. Redhat has **netcfg** for example. You can also do this manually using the **ifconfig** and **route** commands. See the man pages on these or the Net HOWTO for more information and specifics. A quick command line example with bogus IPs:

```
# ifconfig eth0 111.222.333.444 up netmask 255.255.255.0
# route add default gw 111.222.333.1 dev eth0
```

Be sure to add the correct nameservers in `/etc/resolv.conf`.

3.2.2. Bridged/DHCP Configuration

ISPs that have Bridged networks typically use DHCP to assign an IP addresses, and authenticate the user. All distributions come with one or more DHCP clients. **dhcpcd** seems to be the most common. **pump** comes with Redhat based distributions as of Redhat 6.0. The DHCP client will obtain an IP "lease" from the ISP's server as well as other related information: gateway address, DNS servers, and network mask. The lease will be "renewed" at regular intervals according to the ISP's configuration.

You will want the DHCP client started on boot, so use your distribution's means of doing this. There generally is little to configure with DHCP as it is fairly straightforward and easy to use. You may need to tell it which interface to listen on if the NIC is something other than "eth0". You can also start it from the command line to get started. See the respective man pages for more.

Unless you have a static IP, the ISP will need some way to know who you are when you connect. There are two ways this authentication process is accomplished with DHCP. The first and most common method is via the MAC (or hardware) address of the network device. Typically this would be the NIC. The MAC address is a unique identifier and can be found among the boot messages, or with **ifconfig**, and looks something like `00:50:04:C2:19:BC`. You will need to give the ISP the MAC address before your first connection.

The other DHCP authentication method is via an assigned hostname. In this case, the ISP will have provided you with this information. Your DHCP client will need to pass this information to the server in order for you to connect. Both **dhcpcd** and **pump** accept the "−h" command line option for this purpose. See the client's man page, or your distribution's documentation, for specifics.

> **Note:** If your ISP uses MAC address authentication, and you change your network device (e.g. NIC), you will need to register the new address with the ISP or you won't be able to connect.

3.2.3. PPPoE Configuration

PPPoE (PPP over Ethernet) is an alternate way for ISPs to control your connection, and is becoming increasingly popular with ISPs. Setting this up is quite different, and may be a little more work than with static IPs or DHCP above. Recent distro releases are now shipping PPPoE clients. If this is not the case for you, then you will have to download one. Check any Linux archive site like http://freshmeat.net, etc. or look below.

Some of the current GPL PPPoE clients available:

- The Roaring Penguin (rp−pppoe): http://www.roaringpenguin.com/pppoe/, by David F. Skoll. Reportedly very easy to set up, and get started with. This is a popular Linux PPPoE clients due to it's reputation for ease of installation, and is now being bundled with some distributions. rp−pppoe works as a user−mode client on 2.0 and 2.2 kernels, and in kernel−mode on 2.4 kernels.
- PPPoEd: http://www.davin.ottawa.on.ca/pppoe/ by Jamal Hadi Salim is another popular Linux client and is also bundled with some distros. This is a kernel based implementation for 2.2 kernels. A setup script is now included so no patching is required, making installation quick and easy. Also, less CPU intensive than user space alternatives like rp−pppoe (2.0/2.2 kernels).
- PPPoE Redirector: http://www.ecf.toronto.edu/~stras/pppoe.html. This is a redirector which allows the use of PPPoE with pppd−2.3.7 or later. No recompiling of other system components are required. It is meant as an interim solution until the 2.4.x series, which will include kernel support of PPPoE/A. (Does not seem to be under active development at this time.)
- 2.4.x kernels include native PPPoE support. The PPPoE for 2.4 page is http://www.shoshin.uwaterloo.ca/~mostrows and is by Michal Ostrowski, the maintainer for kernel PPPoE.
- EnterNet is a non−GPL'd PPPoE client from NTS, http://www.nts.com, that is being distributed by some ISPs as the Linux client. It does come with source code but the it is not available for free download. (I haven't found anyone that is impressed by this one.)

Depending on which client you have chosen, just follow the `INSTALL` instructions and other documentation included with that package (`README`, `FAQ`, etc.).

Once a PPPoE client connects, your connection should look something like the below example from Roaring Penguin, where "eth0" is connected to the modem:

```
$ route −n

Kernel IP routing table
Destination     Gateway      Genmask         Flags Metric Ref Use Iface
192.168.0.254   *            255.255.255.255 UH    0      0     0 eth1
208.61.124.1    *            255.255.255.255 UH    0      0     0 ppp0
192.168.0.0     *            255.255.255.0   U     0      0     0 eth1
127.0.0.0       *            255.0.0.0       U     0      0     0 lo
default         208.61.124.1 0.0.0.0         UG    0      0     0 ppp0


$ ifconfig

eth0    Link encap:Ethernet  HWaddr 00:A0:CC:33:74:EB
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:297581 errors:0 dropped:0 overruns:0 frame:0
        TX packets:266104 errors:1 dropped:0 overruns:0 carrier:2
        collisions:79 txqueuelen:100
        Interrupt:10 Base address:0x1300

eth1    Link encap:Ethernet  HWaddr 00:A0:CC:33:8E:84
        inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:608075 errors:0 dropped:0 overruns:0 frame:0
        TX packets:578065 errors:0 dropped:0 overruns:0 carrier:0
        collisions:105408 txqueuelen:100
        Interrupt:9 Base address:0x1200

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:3924  Metric:1
        RX packets:1855 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1855 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0

ppp0    Link encap:Point−to−Point Protocol
        inet addr:208.61.124.28  P−t−P:208.61.124.1  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
        RX packets:297579 errors:0 dropped:0 overruns:0 frame:0
        TX packets:266102 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:10
```

> **Note:** PPPoE adds some extra overhead to the ethernet frames and the correct setting for the ppp0 interface MTU is 1492. If the MTU is set too high, it may cause failure of some web pages to load properly, and possibly other annoying problems. You may need to also set the MTU for interfaces on any masqueraded LAN connections MTU to 1452. This does not apply to PPPoA, or bridged configurations, just PPPoE!

Actually, for PPPoE the real setting should be at least 8 bytes less (the extra PPPoE protocol overhead) than any interface between you and the ultimate destination. All routers normally would be set to 1500, thus 1492 is correct from your end. But, it may happen that somewhere a router is configured at a lower setting, and this can cause problems, especially with web pages loading, and other traffic failures. The way to test this is to keep dropping the MTU until things work.

3.2.4. PPPoA

PPPoA (PPP over ATM) is a cleaner solution than PPPoE since most of the work is done in hardware, and since the raw DSL traffic is ATM. There is no user space client necessary to manage the connection as with PPPoE, and the additional ethernet protocol layer is not required. Authentication is still the same: user id and password to connect, but the mechanics are different since no ethernet encapsulation takes place.

PPPoA is either done completely in hardware or is implemented as a device specific driver. There is no such thing as a generic PPPoA software client like there is for PPPoE. There is an ATM patch for 2.2 kernels, support for ATM in the 2.4.x kernel, and a project based on the Efficient Networks 3010, as well as other ATM cards. The ATM on Linux homepage is here: http://lrcwww.epfl.ch/linux−atm/. And even more info is at http://www.sfgoth.com/~mitch/linux/atm/pppoatm/ from the kernel developer of this project. Existing PPPoA implementations are hardware/driver based, and Linux PPPoA modem drivers are scarce as hen's teeth at this time. The above modem does not seem to be available through normal retail channels. This may be a problem, if this is the only protocol an ISP delivers, and an external modem that supports PPPoA is not available.

If PPPoA is your ISP's only option, you might consider one of the router/modems that can handle PPPoA connections, and let the hardware handle everything.

3.2.5. PPTP/PPPoA with Alcatel Ethernet Modems

Alcatel SpeedTouch Home ethernet modems (supersedes the Alcatel 1000) support both bridged and PPPoA connections. The modem itself handles the PPPoA protocol internally. When in PPTP/PPPoA mode (as opposed to RFC1483 bridging mode), Linux will connect to the modem via PPTP (MS VPN). The Linux PPTP homepage is http://cag.lcs.mit.edu/~cananian/Projects/PPTP/, and works well with this modem. In addition to installing pptp, your kernel must also have support for PPP.

The modem has internal configuration pages than can be reached by pointing a browser to the default IP address of http://10.0.0.138. (You will of course have to have your NIC set up for a 10.0.0.0 network with similar IP such as 10.0.0.1, in order to reach the modem's configuration pages.) For PPPoA, the connection type is 'PPTP'. You will have to get the other settings from your provider if the defaults do not work. Settings such as 'VPI/VCI' and 'encapsulation' can vary from provider to provider. Of course, if the modem is coming from your provider, all this should be already configured.

The next step is to configure **pptp**, which is done by configuring the **pppd** files `/etc/ppp/pap−secrets` (or `chap−secrets`) and `/etc/ppp/options`. This is where the username and password is entered. For example:

`/etc/ppp/pap−secrets`:

```
# client secret server IP address
login@isp.com  *    my_password_here    *
```

and `/etc/ppp/options`:

```
name "login@isp.com"
noauth
noipdefault
defaultroute
```

Once everything is configured properly, it should be just a matter of starting pptp, pointing it to the modem's address:

```
#pptp 10.0.0.138
```

> **Note:** Alcatel supplies many sub−models of these modems. These features may not be available on all models, or may be altered from the defaults. This is something to be aware of, if buying a used modem.
>
> This modem only supports one concurrent PPTP connection.

3.2.6. Modem/Router Configuration

Some ISPs are providing "routers" as the connection device. Essentially these are mini routers with built in modems. These are all ethernet based devices too, so Linux should be good to go here as well. Again, a compatible, working NIC should be all that is required to make this work.

A "router" has many advantages. The better ones can handle the connection management, IP encapsulation, and authentication, as well as providing a means of segregating your LAN from outside traffic, and possibly other features too. In short they can do it all. One big advantage is that they can handle whatever protocols your ISP requires in order to connect.

If the ISP is requiring PPPoX, then this makes life a little easier since you will not have to install or configure any additional software just to use their network. The modem's firmware will handle this. The downside is that most of these do not have the flexibility of a Linux router, or other software solution. Of course, you could set up a Linux router behind the router, and have the best of both worlds. The ones with more and better features are also going to cost significantly more.

While the physical installation of a router is very similar to the modem installation (see above), the router configuration itself is different since your first "hop" will be the router's interface and not the ISP's gateway. Routers will actually have two interfaces −− one that you connect to from the LAN side, and one that connects to your ISP on the WAN side. Your point of exposure here is the WAN interface of the router.

The router will also have a pre−configured, private IP address that you will connect to from the LAN side. This will be your gateway. The public IP address will be assigned to the WAN side interface. Typically these devices also act as DHCP servers for the LAN side as well. So possibly all you have to do is to start a DHCP client such as **dhcpcd** or **pump** (Redhat based distros) to get up and running. Just make sure the modem/router is syncing first. The appropriate steps and configuration should be in the owner's manual, or available from your provider.

If you are a PPPoX customer, and the router is handling this part of the connection, then you will have to configure at least your user id and password before connecting. If a Bridged/DHCP customer, you should just have to activate DHCP on the router, and possibly register the MAC (hardware address) of the router with your provider. Some routers have "MAC cloning" which means that they will report the MAC address of the attached NIC. If static IP, then you will have to configure this as well.

If you need to access the router directly, you will need to know the manufacturer's default setting for its IP address. See the owner's manual, or ask your provider. You will then have to set your NIC's interface to the same network as the router. For instance, if the router has an IP of 10.0.0.1, set your interface's address to 10.0.0.2 (typically eth0), and netmask to 255.0.0.0.

```
# ifconfig eth0 10.0.0.2 up netmask 255.0.0.0
# route add −net 10.0.0.0
$ ping 10.0.0.1
```

If everything is in working order, the router should respond to pings. How to configure this permanently will vary from distro to distro. So check your distribution's documentation. Now you should be able to ping the modem/router, and, if all is well, beyond. Then use telnet or a web browser to do any further configuration of the router.

Even if the ISP is not offering any router options, there are quite a few available from third party manufacturers such as Netgear, Linksys, Cisco, Zyxel, Cayman, Alcatel and others. These will have all the features already mentioned and maybe more. Just make sure it matches your provider's DSL. This is one good way around the PPPoX bugaboo.

Caution

Some manufacturers may be marketing these as having "firewall" capabilities. In some cases, this amounts to nothing more than basic NAT (Network Address Translation or masquerading). Not a full, true firewall by any means. Be sure to read the fine print before buying and make sure you know how much real firewalling is included.

3.3. Connect

Everything should be in place now. You probably have already tested your connection. You should be seeing ping roundtrip times of 10−75 ms to the ISP's gateway. If something has gone wrong, and you cannot connect, either retrace the above steps, or see the Troubleshooting Section below.

4. Securing Your Connection

This section is intended for those who have not previously dealt with the security implications of having a full−time Internet connection. Or may not understand some of the basic concepts of security. This is meant to be just a quick overview, not a comprehensive examination of all the issues! Just enough to give you a quick push in the right direction. Please see the Links section for sites with more details. Also, your distribution surely has plenty of good information as well.

4.1. Security Quick−start

Before going on−line full−time, do not underestimate the need for securing your connection. You will have two things that mischief makers and crackers of the world are looking for: bandwidth, and a Unix−like OS. You instantly become an inviting target. It is just a matter of time before someone comes knocking. Possibly a very short time. A quick start:

- Turn off any daemons and services that aren't absolutely essential, and can be accessed from outside. You can't get compromised through a port that isn't open. Use **ps** and **netstat** to see what services are running. (See man pages for specifics). Do you really need **named**, **sendmail**, **telnet**, **ftp** running and accessible to one and all? If not sure, then they should not be running. Then take whatever steps necessary to make sure they don't start again on the next boot. See your distribution's documentation on this.

  Many distributions start some well known services by default. You may not have done anything yourself explicitly to start these. And may not even realize these are indeed running. But it is up to you to know what is running, and how safe it is. Don't rely on a "default" installation of any distribution to do this for you, or to be secure. Chances are it isn't.

- If you decide some services are essential, make sure you are running the most current version. Exploits are found, and then get fixed quickly. Don't get caught with your pants down. A full−time connection makes staying updated very easy −− and very important. Check with your distribution to see what new packages are available. Then stay in touch. If they have a security mailing list, get on it.
- Set up a firewall to limit access, and log connection attempts. This will be different depending on which kernel series you are using: **ipfwadm** for 2.0, **ipchains** for 2.2, and **iptables** for 2.4. See these HOWTOs for details:
-
    - Firewall HOWTO
    - Security HOWTO
    - IPCHAINS HOWTO
    - IP Masquerade HOWTO

  Additional references are in the Links Section below.

- Take passwords seriously, using non−dictionary "words". Use shadow passwords (this should be a standard feature of newer distributions). Do not allow remote root logins. See the Security HOWTO for more details and ideas.
- Use **ssh**, or **OpenSSH**, instead of **telnet**.

4.2. Which Ports?

There are plenty of references around on how to setup firewalls with **ipfwadm** or **ipchains**, but a little harder to find just which ports *really need* to be open. If you are not sure of the answer to this question, then the answer is "as few as possible"! Basic rule #1, your computer cannot be broken into through a port that is not open. A port can't be open if nothing is listening on that port. In other words, if no service or daemon is running that uses that port, the port is closed and inaccessible.

There are 131,072 ports (TCP and UDP) available for use on Linux and these fall into several categories. The ones we are most concerned about are the "privileged" ports, which are those below port 1024. This is where most public services run, like SMTP on port 25, HTTP on port 80, named on port 53, etc. These services are where most vulnerabilities are on Linux. These are the ports that servers accept outside connections on. If you are running the **telnetd** daemon, it will "listen" for connections on port 23. (Actually if spawned by **inetd**, it may be **inetd** that is doing the listening.)

But, if you want to use a telnet *client* to connect to someone else's telnet server, you do not need to be running the **telnetd** daemon on your own system. You just need to start the client program named **telnet** (different animal). Same is true for ftp and other services. These daemons only need to be run for excepting connections to your system from an outside source.

Unless you have a good reason for doing so, and know what you are doing, then you should not be running such publicly accessible services. In fact, you could probably survive quite nicely with all TCP and UDP ports below 1024 closed down, or at least not visible to outside connections (i.e. blocked via a firewall). A few possible exceptions:

It is relatively safe, and in some cases alright, to run **identd** (port 113). Many mail and irc servers aren't happy without **identd** there. This is the one possible exception to the "nothing below 1024" rule of thumb. Newer versions are reasonably secure.

Mail is a little more complicated. The only reason to have a publicly accessible SMTP (port 25) server is if you are hosting your own mail server and receiving direct incoming mail connections. If you are retrieving mail via POP3 from your ISP, this port does not need to be open to the world. Such mail does not come in via port 25. It comes in through a higher, randomly assigned port.

But −− it may still be convenient to have a mail daemon like sendmail, qmail, or postfix running to handle local mail delivery. In fact, this is a common default set up. You can get around this by firewalling off SMTP (port 25) from the outside world, or using another method to sort and deliver local mail. One way is to use procmail in conjunction with fetchmail: `fetchmail −m /usr/bin/procmail −d hal` will do this without having to run sendmail or other mail daemons in daemon mode. These programs are still capable of sending mail in non−daemon mode.

Older versions of BIND required that port 53 UDP be opened (TCP required for zone transfers). BIND is only appropriate if you are running a nameserver (or caching nameserver). The solution here is to upgrade to a newer version that uses unprivileged ports by default.

It is probably safe to run a web server if you want to. Most vulnerabilities there are through CGI. Just keep the web server package updated.

OK, enough exceptions. Shut down, or firewall off, *everything else* below port 1024.

Those ports above 1023 are known as "non−privileged" ports. These are mostly used for return connections that you have initiated to someone else's server. For instance, if you telnet to someone else, you connect to their port 23. The return data comes back to you on a randomly assigned port above 1023. These are mostly safe, and should as a rule be left alone. The only exceptions are where there are indeed services running on these ports. X Windows runs on ports 6000−6009 for instance. If you are running a font server, it may be on port 7100. Any servers running on these non−privileged ports, should be firewalled too.

4.3. inetd

Let's take a quick look at **inetd**, since it starts many services. **Inetd** is a "super" daemon. It is called this because it controls the starting of other daemons. Telnet, ftp, rsh, identd and pop3 are some of the server daemons commonly controlled by **inetd**. You may not see **telnetd** running when you use **ps** or **netstat** unless **inetd** is configured to start **telnetd**, *and* someone is actually connected to **telnetd** at the time. So it may not be so obvious which of these servers can be accessed from outside. These sub−services are controlled by the configuration file `/etc/inetd.conf`. Open this with your favorite text editor, and put a "#" character in front of any service you don't want running. A brief excerpt:

```
 #ftp    stream tcp nowait root /usr/sbin/tcpd in.wuftpd −l −a −L −i −o
 #telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
 #
 # Shell, login, exec, comsat and talk are BSD protocols.
 #
 #shell stream  tcp   nowait  root   /usr/sbin/tcpd  in.rshd
 #login stream  tcp   nowait  root   /usr/sbin/tcpd  in.rlogind
 #exec  stream  tcp   nowait  root   /usr/sbin/tcpd  in.rexecd
 #comsat dgram  udp   wait    root   /usr/sbin/tcpd  in.comsat
 #talk   dgram  udp   wait    root   /usr/sbin/tcpd  in.talkd
 #ntalk  dgram  udp   wait    root   /usr/sbin/tcpd  in.ntalkd

<snip>

 auth   stream  tcp   nowait  nobody /usr/sbin/in.identd in.identd −l −e −o
```

This will prevent them from running. **identd** may be started from this file, and would be safe to leave uncommented. Then re−initialize: **inetd**:

```
# kill −HUP `pidof inetd`
```

Note those are "backquotes". For more on fine tuning inetd access via "tcpwrappers", see the hosts_access and tcpd man pages.

xinetd is a similar daemon, and you may have this instead of **inetd**. It does pretty much the same thing, so the same principles apply. The configuration is a little different however. Each service is either configured in a central configuration file (typically `/etc/xinetd.conf`), or in a subdirectory (typically `/xinetd.d/*`.)

```
 # description: The telnet server serves telnet sessions; it uses
 # unencrypted username/password pairs for authentication.

 service telnet
 {
disable     = yes
flags       = REUSE
socket_type = stream
wait        = no
user        = root
server      = /usr/sbin/in.telnetd
log_on_failure += USERID
 }
```

To turn off a service, set "disable" to "yes", and restart xinetd.

5. Performance Tuning and Troubleshooting

5.1. Tuning

OK, now we are up and running, and we want to be running at warp factor nine. No such thing as too fast, right?

Linux networking is pretty robust, even a default installation with no "tuning". You may well not need to do anything else. But if your connection is not performing up to what you think it should be, then possibly there is a problem somewhere. This is may be a more worthwhile approach than the pursuit of any magical "tweak".

A *very* rough guideline on what you might reasonably expect as a maximum sync rate, based on distance from DSLAM/CO:

```
    0−12 K ft  (0−3.6 km)   − 2000 Kbps or more (8100 max for ADSL)
  12−16 K ft (3.6−4.6 km) − 1500 Kbps -> 1000 Kbps
  16−18 K ft (4.6−5.4 km) − 1200 Kbps ->  512 Kbps
  18−?? K ft (5.4−?? km)  −   512 Kbps ->   128 Kbps or less :(
```

There are many conceivable factors that could effect this one way or the other. Newer generations of DSL will surely improve this, as will related technologies like repeaters.

You will loose 10−20% of the modem's attainable sync rate to networking overheads (TCP, ATM, ethernet). So a 1500 Kbps connection, is only going to realize about 1100−1300 Kbps or so of real world throughput. No tweaking is going to change the built−in protocol overheads. Also, if your service is capped at a lesser speed by your provider, then you can't get above that speed no matter what. *AND* −− that there are numerous variables that can effect your loop/signal quality, and subsequently your speed. Some of these may be beyond your control.

5.1.1. TCP Receive Window

For many of us, a default Linux installation is going to give something close to optimum performance. Windows 9x users often get a big boost by increasing their TCP Receive Window (RWIN). But this is because it is too small to start with. This is just not the case with Linux where the default value is 32KB.

The exception here is if you have to routinely deal with a high latency connection. For instance, if your provider has a satellite uplink that is consistently adding unusual latency (250ms or greater?). Then a larger TCP Window will likely help. For more on TCP Receive Window and related issues, look at http://www.psc.edu/networking/perf_tune.html.

The Receive Window is a buffer that helps control the flow of data. If set too low, it can be a bottleneck and restrict throughput. The optimum value for this depends completely on bandwidth and latency. Latency being what you would find as average roundtrip time (RTT) based on *your* typical destinations. This can be determined with **ping**. For example, the Linux default of 32KB is acceptable up to speeds of 2 Mbps and typical latency of 125ms or so, or 1.0 Mbps and latency of 250ms. Setting this value too high can also adversely effect throughput, so don't over do it.

An example courtesy of Juha Saarinen of New Zealand:

> The commonly used formula for working out the the tcp buffer is the "bandwidth delay product" one:
>
> Buffer size = Bandwidth (bits/s) * RTT (seconds)
>
> In my case, I have roughly 8Mbps downstream, but the ATM network can only support ~3.5Mbps sustained. I'm far away from the rest of the world, so to squeeze in a sufficient amount of 1,500 byte packets, with average RTTs of 250ms, I should probably have a buffer of (3,500,000/8)*.25 = 106KB. (I've got 128KB at the moment, which works fine.)

The Receive Window can be dynamically set in the /proc filesystem. This requires entering a value that is twice the desired buffer size:

```
 #echo 262144 > /proc/sys/net/core/rmem_default
 #echo 262144 > /proc/sys/net/core/rmem_max
```

The above example actually sets the value to 128K. The Send Window can also be set, but is not as likely to be a limiting factor on DSL connections as the Receive Window:

```
 #echo 262144 > /proc/sys/net/core/wmem_default
 #echo 262144 > /proc/sys/net/core/wmem_max
```

These values can also be set using the **sysctl** command. See the man page.

Other suggested kernel options for those who want to squeeze every last bit out of that copper (selected entries only):

```
# sysctl -a
net.ipv4.tcp_rfc1337 = 1
net.ipv4.ip_no_pmtu_disc = 0
net.ipv4.tcp_sack = 1
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_timestamps = 1
```

5.1.2. Interleaving

"Interleaving" is an error control mechanism of ADSL with DMT line encoding. DMT is now the standard for ADSL, and is by far and away the most prevalent form of ADSL. Interleaving buffers the raw data and corrects errors on the fly at the DSLAM. This can significantly help marginal loops that may be prone to errors. The downside is that this buffering also adds significant latency to the connection. So for those with reasonable quality lines, interleaving is of no real benefit, and may actually slow things down unnecessarily.

Interleaving is an adjustable parameter and can be turned on or off by the telco. Most telcos seem to like to have this on by default, since it probably reduces tech support calls in those cases where it does help stabilize a line. But everyone else is paying a price.

How to know if your line is interleaved or not, and how to change it? Good question. Generally speaking, if your first hop or two on a traceroute is less than 25ms or so, you can pretty much figure that interleaving is off. But there may be other factors such as how far away those hops actually are. The only real way to know is to talk to someone at the telco. This may prove easier said than done.

"FastPath" DMT is synonymous with "interleaving off". Again, this *only* applies to ADSL/DMT.

5.2. Installation Problems

Read this section, if you have no sync at all or are completely unable to connect. See your modem's owner's manual for interpreting the modem's LEDs. (Many will show a solid red (or orange) light if not in sync.)

5.2.1. No sync

The modem sync LED has never been green.

- If doing a self−install, the DSL jack may be wired wrong, or the splitter may be wired wrong. Also, the modem may be wired differently than standard telco POTS devices. See above.
- Is the modem Linux compatible? If ethernet interfaced, this should not be a problem. But PCI or USB modems may require drivers just to achieve sync. This could be a show stopper since most PCI and USB modems are not Linux compatible.
- Call your provider and make sure the line was provisioned. It is always possible someone dropped the ball. They may even be able to run a remote test on your line just to verify. There is a also remote possibility that the DSLAM is down. They should know this as well.
- Disconnect the modem power cord and disconnect the DSL cord from the wall jack. Plug it into the test jack inside the NID (outside phone box), and run an extension cord if necessary for power. Temporarily disconnect the wiring to the inside phone circuit. This should effectively bypass any inside wiring and environmental issues. The ethernet cable to the NIC does not need to be connected to run this test (true *only* for ethernet modems). The modem will sync fine without it. (Easier said than done, I know.) But if possible, move enough of your system where you can view the modem's diagnostics (if available) and get the sync rate. If this works, there is probably something wired incorrectly inside, or a short in a connection somewhere, or there is severe electrical interference on the DSL line. Double check the splitter and wall jack connections. If a splitterless installation, look for bad wiring, bad (e.g. corroded) connections on *all* jacks, bad splices, or defective microfilters!

  If no sync on the above test, either the line was not readied, the modem is defective, or the DSLAM is down. Note that PCI and USB modems will need to load drivers before syncing, and thus make this test a little more complicated.
- If you installed microfilters, remove these temporarily and unplug *all* telco devices, such as fax machines, etc. Possibly a mircofilter is defective and shorting out the line.

5.2.2. Network Card (NIC) Problems

Symptoms here are: NIC is not recognized, modules won't load, or **ifconfig** shows the interface is not up, or is generating lots of errors, etc.

- Turn off Plug 'n Pray in BIOS. This may be labeled as "non−Microsoft OS" or similar. A sometimes symptom here is that the NIC is assigned IRQ 0. Or there may be an error message like "resource temporarily unavailable".
- Check for IRQ conflicts with **cat /proc/interrupts**. If the NIC is sharing an IRQ, try moving cards around in slots, or tinker with BIOS IRQ settings. If an ISA card, you may need to get the setup utility from the manufacturer and use it to set IRQ, etc. This may require booting to DOS.
- Possibly the wrong module is being loaded. Look through the kernel source documentation in `/usr/src/linux/Documentation/*` for your card or chipset. Also, for comments and update information in `/usr/src/linux/drivers/net/*.c` for your respective chipset. It is worth noting that there is more than one module for some card types. This seems to be true of tulip and 3Com cards. Check boot messages or use **lspci −v** to see how the kernel is identifying your card. You can use **insmod**, **rmmod**, and **modprobe** to test different modules. See the respective man pages for more information.
- Check the manufacturer's web site for Linux documentation. Or look at Donald Becker's informative site at http://www.scyld.com/network/.
- Some Linux NIC drivers reportedly work better as non−modular. In other words, compile them into the kernel instead of as a module.
- It is also possible that the card is bad, or the drivers just aren't up to snuff. Try another card. And you don't need an expensive, high quality card necessarily either.

5.2.3. IP Connection Problems

Read this section if you are sure the modem is syncing, the NIC is recognized and seems to be working properly, client software is installed and running without error, but the connection to the ISP fails. Verify the modem is indeed syncing by the LED(s). An IP connection failure may be evidenced by **ifconfig** not showing an active eth0 interface (or ppp0 for PPPoX), or **pinging** gateway and other destinations generates '`network unreachable`' or similar errors.

- Make sure you know which protocol your ISP is using. Are they using DHCP? PPPoX? It is critical that you have this right. You may have to ask tech support.
- If you are using DHCP, does the ISP require MAC address authentication, and if so, do they have the right address? Did they or you typo it? If the ISP requires hostname authentication, is your DHCP client passing the required hostname? This is done with the "− h" command line option.
- Look at `/var/log/messages` and see if any useful clues are there. Also, run **tcpdump** while trying to initiate the connection. **tcpdump** output is fairly cryptic, but you should be able to determine if there is any response at all.
- If PPPoX, is the ISP using `username` as an id, or `username@isp.com`?
- Try pinging the default gateway's address. Get this with '**route −n**'. If you can **ping** by IP address (i.e. 111.222.333.444), but not by hostname, then likely nameservers are not correctly setup in `/etc/resolv.conf`.
- For PPPoE, let the PPPoE client bring up the ethernet interface. Do not have it come up on boot. Make sure there is no existing default route before starting PPPoE. For rp−pppoe, David Skoll recommends that `/etc/ppp/options` be left empty.
- If running a firewall (e.g. with ipchains), try temporarily taking it down. Possibly this is misconfigured, and not allowing packets through.
- If the modem was purchased from a source other than your ISP, it may the wrong kind of modem. SDSL needs an SDSL modem, for instance. Also, for ADSL there are CAP and DMT encodings, and these are incompatible with each other.

The modem may need to be configured for your ISP's service. All modems have configurations for VCI, VPI, encapsulation, etc. Call tech support for this information. Modem configuration is usually done by either telnetting or web browsing to the modem's IP address.

5.3. Sync Problems

Read this section if you have had a working connection, but now have lost sync, are intermittently losing sync, your sync rate has dropped significantly, or are getting a "sync/no surf" condition. (Better quality modems will have a way to report sync rate, usually via telnet or a web browser interface. See the owner's manual.)

A loss of sync indicates a problem with the DSLAM, your line (inside or outside) or your modem. DSLAMs typically have "shelves" with "cards". Alcatel DSLAM cards, just for instance, have a capacity of four connections each. If the card goes bad, at most four customers are effected. The point being that sync loss outages can be very isolated. Unlike network outages that tend to effect large numbers of users. Sync outages are a telco problem, not an ISP problem. If your service agreement is with the ISP, you will need to contact them, who will in turn contact the telco.

Degraded sync rates, and disruption of the DSL signal, can cause various problems. Obviously, you will never get your maximum throughput under these conditions. But, the symptoms are not always obvious as to whether the problem is on your end or the provider's.

For instance, a poor inside wire connection may result in retransmissions of packets that have been dropped. This can really reduce throughput and slow a connection down. It is tempting to think of packet loss as a traditional networking problem, but with DSL it is possible to be the result of a bad line, impaired signal, or even the modem itself.

Some things to try:

- Power cycle the modem. Turn off the power button/switch, and physically unplug the cable to the wall jack for 30 seconds or so. Turn back on, and re−attach to the wall jack. This will force a resync. Unfortunately, the only way to power down a PCI modem, is to reboot. This may fix a "sync/no surf" condition that is caused by the modem, and maybe other conditions too.
- See the above section on moving the modem lock, stock and barrel to the NID and thus bypassing all inside wiring. If the situation is improved there, then the problem is inside somewhere. If not, it is a telco problem.
- RFI Bear−hunt: The DSL signal is fragile, there are a number of things that can degrade it. RFI, or Radio Frequency Interference, from sources in and around the home/office is one common source of reduced signal strength, and/or intermittent sync loss and/or low sync rates. Our test tool here is simply a portable AM radio. Tune it to any channel where you can get clear reception −− it makes no difference where. The AM radio will pick up RFI that is in the same frequency range as the DSL signal. It will sound like "frying bacon" type static. Put it against your computer's power supply. You should hear some static. Move it away and the static should fade pretty quickly. This will give you an idea of what RFI sounds like. A decent quality power supply should produce only weak RFI −− probably not enough to cause a problem. Use the radio like a Geiger counter and move it around your modem and DSL line. If you hear static, follow it to the source. Things to be suspicious of: power supplies, transformers, ballasts, electric motors, dimmer switches, high intensity lighting. Moving the modem, or rerouting cables is sometimes enough. Keeping the line between the modem and the wall jack as short as possible is a good idea too.

- Chronic sync problems are often due to a line problem somewhere. Sometimes it is something as simple as a bad splice, and easily remedied if it can be found. Most such conditions can be isolated by a good telco tech. Check with your provider, and politely harass them if you have to. If you get the run−around, ask to go over their heads.
- If you are near the distance limits of DSL, and having off and on sync problems, try the "Homerun" installation. See above. This can be effective in improving marginal signal/sync conditions.
- If using a surge protector, try it without the surge protector. Some may interfere with the DSL signal.

Another possibility is a nearby AM radio station, or bandit ham radio operator that are disrupting the DSL signal since they operate in a similar frequency range. These may only cause problems at certain times of day, like when the station boosts its signal at night. A good telco DSL tech may be able to help minimize the impact of this. YMMV.

5.4. Network and Throughput Problems

Read this section if your connection is up, but are having throughput problems. In other words, your speed isn't what it should be based on your bit rate plan, and your distance from the CO. "Network" here is the WAN −− the ISP's gateway and local subnet/backbone, etc. Remember that a marginal line can cause a reduced sync rate, and this will impact throughput. See above.

The two factors we will be looking for are "latency" and "packet loss". Both are pretty easy to track down with the standard networking tools **ping** and **traceroute**. If either of these occur in our path, they will impact performance. Latency means "responsiveness" or "lag time". Actually what we are interested in is abnormally high latency, since there is always some latency. Packet loss is when a packet of data gets dropped somewhere along the way. TCP/IP will know it's been "lost", and there will be a retransmission of the lost data. Enough of this can really slow things down. Ideally packet loss should be 0%.

What we really need to be concerned about is that part of the WAN route that we routinely traverse. If you do a traceroute to several different sites, you will probably see that the first few "hops" tend to be the same. These are your ISP's local backbone, and your ISP's upstream provider's gateway. Any problem with any of this, and it will effect everywhere you go and everything you do.

We can start looking for packet loss and latency by pinging two or three different sites, hopefully in at least a couple of different directions. We will be looking for packet loss and/or unusually high latency.

```
$ ping −c 12 −n www.linuxdoc.org
PING www.linuxdoc.org (152.19.254.81) : 56(84) bytes of data.
64 bytes from 152.19.254.81: icmp_seq=0 ttl=242 time=62.1 ms
64 bytes from 152.19.254.81: icmp_seq=1 ttl=242 time=60.8 ms
64 bytes from 152.19.254.81: icmp_seq=2 ttl=242 time=59.9 ms
64 bytes from 152.19.254.81: icmp_seq=3 ttl=242 time=61.8 ms
64 bytes from 152.19.254.81: icmp_seq=4 ttl=242 time=64.1 ms
64 bytes from 152.19.254.81: icmp_seq=5 ttl=242 time=62.8 ms
64 bytes from 152.19.254.81: icmp_seq=6 ttl=242 time=62.6 ms
64 bytes from 152.19.254.81: icmp_seq=7 ttl=242 time=60.3 ms
64 bytes from 152.19.254.81: icmp_seq=8 ttl=242 time=61.1 ms
64 bytes from 152.19.254.81: icmp_seq=9 ttl=242 time=60.9 ms
64 bytes from 152.19.254.81: icmp_seq=10 ttl=242 time=62.4 ms
64 bytes from 152.19.254.81: icmp_seq=11 ttl=242 time=63.0 ms

−−− www.linuxdoc.org ping statistics −−−
```

```
12 packets transmitted, 12 packets received, 0% packet loss
round−trip min/avg/max = 59.9/61.8/64.1 ms
```

The above example is pretty normal from here. (You probably have a very different route to this site, and your results may thus be quite different.) Apparently no serious underlying problems that would slow me down. The below example reveals a problem:

```
$ ping −c 20 −n www.debian.org

PING www.debian.org (198.186.203.20) : 56(84) bytes of data.
64 bytes from 198.186.203.20: icmp_seq=0 ttl=241 time=404.9 ms
64 bytes from 198.186.203.20: icmp_seq=1 ttl=241 time=394.9 ms
64 bytes from 198.186.203.20: icmp_seq=2 ttl=241 time=402.1 ms
64 bytes from 198.186.203.20: icmp_seq=4 ttl=241 time=2870.3 ms
64 bytes from 198.186.203.20: icmp_seq=7 ttl=241 time=126.9 ms
64 bytes from 198.186.203.20: icmp_seq=12 ttl=241 time=88.3 ms
64 bytes from 198.186.203.20: icmp_seq=13 ttl=241 time=87.9 ms
64 bytes from 198.186.203.20: icmp_seq=14 ttl=241 time=87.7 ms
64 bytes from 198.186.203.20: icmp_seq=15 ttl=241 time=85.0 ms
64 bytes from 198.186.203.20: icmp_seq=16 ttl=241 time=84.5 ms
64 bytes from 198.186.203.20: icmp_seq=17 ttl=241 time=90.7 ms
64 bytes from 198.186.203.20: icmp_seq=18 ttl=241 time=87.3 ms
64 bytes from 198.186.203.20: icmp_seq=19 ttl=241 time=87.6 ms

−−− www.debian.org ping statistics −−−
20 packets transmitted, 13 packets received, 35% packet loss
round−trip min/avg/max = 84.5/376.7/2870.3 ms
```

High packet loss at 35%, and some really slow roundtrip times in there as well. A little digging on this showed that it was a backbone router 13 hops into the traceroute that was the problem. While making this site really slow from here, it would only effect those routes that happen to hit that same router. Now what would really hurt us is if something similar happens with a router that we tend to go through consistently. Like our gateway, or maybe the second hop router too. Find these with **traceroute**, by just picking a random site:

```
$ traceroute www.bellsouth.net

traceroute to bellsouth.net (192.223.22.134), 30 hops max, 38 byte packets
 1  adsl−78−196−1.sdf.bellsouth.net (216.78.196.1)  14.86ms  7.96ms 12.59ms
 2  205.152.133.65 (205.152.133.65)                  7.90ms  8.12ms  7.73ms
 3  205.152.133.248 (205.152.133.248)                8.99ms  8.52ms  8.17ms
 4  Hssi4−1−0.GW1.IND1.ALTER.NET (157.130.100.153)  11.36ms 11.48ms 11.72ms
 5  125.ATM3−0.XR2.CHI4.ALTER.NET (146.188.208.106) 14.46ms 14.23ms 14.40ms
 6  194.at−1−0−0.TR2.CHI2.ALTER.NET (152.63.65.66)  16.48ms 15.69ms 16.37ms
 7  126.at−5−1−0.TR2.ATL5.ALTER.NET (152.63.0.213)  65.66ms 66.18ms 66.39ms
 8  296.ATM6−0.XR2.ATL1.ALTER.NET (152.63.81.37)    66.86ms 66.42ms 66.40ms
 9  194.ATM8−0.GW1.ATL3.ALTER.NET (146.188.233.53)  67.87ms 68.69ms 69.63ms
10  IMVI−gw.customer.ALTER.NET (157.130.69.202)     69.88ms 69.25ms 69.35ms
11  www.bellsouth.net (192.223.22.134)              68.74ms 69.06ms 68.05ms
```

The first hop is the gateway. In fact, for me the first two hops are *always* the same, and the first three or four are often the same. So a problem with any of these may cause a problem anywhere I go. (The specifics of your own situation may be a little different than this example.) A "normal" gateway ping (normal for me!):

```
$ ping −c 12 −n 216.78.196.1

PING 216.78.196.1 (216.78.196.1) : 56(84) bytes of data.
64 bytes from 216.78.196.1: icmp_seq=0 ttl=64 time=14.6 ms
```

```
64 bytes from 216.78.196.1: icmp_seq=1 ttl=64 time=15.4 ms
64 bytes from 216.78.196.1: icmp_seq=2 ttl=64 time=15.0 ms
64 bytes from 216.78.196.1: icmp_seq=3 ttl=64 time=15.2 ms
64 bytes from 216.78.196.1: icmp_seq=4 ttl=64 time=14.9 ms
64 bytes from 216.78.196.1: icmp_seq=5 ttl=64 time=15.3 ms
64 bytes from 216.78.196.1: icmp_seq=6 ttl=64 time=15.4 ms
64 bytes from 216.78.196.1: icmp_seq=7 ttl=64 time=15.0 ms
64 bytes from 216.78.196.1: icmp_seq=8 ttl=64 time=14.7 ms
64 bytes from 216.78.196.1: icmp_seq=9 ttl=64 time=14.9 ms
64 bytes from 216.78.196.1: icmp_seq=10 ttl=64 time=16.2 ms
64 bytes from 216.78.196.1: icmp_seq=11 ttl=64 time=14.8 ms

−−− 216.78.196.1 ping statistics −−−
12 packets transmitted, 12 packets received, 0% packet loss
round−trip min/avg/max = 14.6/15.1/16.2 ms
```

And a problem with the same gateway on a different day:

```
$ ping  −c 12 −n 216.78.196.1

PING 216.78.196.1 (216.78.196.1) : 56(84) bytes of data.
64 bytes from 216.78.196.1: icmp_seq=0 ttl=64 time=20.5 ms
64 bytes from 216.78.196.1: icmp_seq=3 ttl=64 time=22.0 ms
64 bytes from 216.78.196.1: icmp_seq=4 ttl=64 time=21.8 ms
64 bytes from 216.78.196.1: icmp_seq=6 ttl=64 time=32.0 ms
64 bytes from 216.78.196.1: icmp_seq=8 ttl=64 time=21.7 ms
64 bytes from 216.78.196.1: icmp_seq=9 ttl=64 time=42.0 ms
64 bytes from 216.78.196.1: icmp_seq=10 ttl=64 time=26.8 ms

−−− adsl−78−196−1.sdf.bellsouth.net ping statistics −−−
12 packets transmitted, 7 packets received, 41% packet loss
round−trip min/avg/max = 20.5/25.6/42.0 ms
```

41% packet loss is very high, to the point where many services, like HTTP, come to a screeching halt. Those services that were working, were working very, very slowly.

It's a little tempting on this last real−life example to think this gateway router is acting up. But, as it turned out, this was the result of a problem in the DSLAM/ATM segment of the telco's network. So any first hop problem with packet loss or high latency, may actually be the result of something occurring before the first hop. We just don't have the tools to isolate where it is starting well enough. Packet loss can be a telco problem, just as much as an ISP/NSP problem.

It is also quite possible for the modem itself to cause packet loss. The fix here is to power cycle the modem, and resync by unplugging the DSL connection for 30 seconds or so. In fact, any part of the connection can be a source of packet loss −− modem, DSLAM, ATM network, etc.

If you do find a problem within your ISP's network, it's time to report the problem to tech support.

5.4.1. Miscellaneous Network Problems

Some odds and ends:

- *Some Web pages won't load.* For PPPoX users, the MTU value could be too high. The correct ppp0 device setting should be 1492, but actually it needs to be 8 bytes less than any router you pass through on the way to the site. If a router somewhere is misconfigured, you could have problems. Try experimenting with lower MTU values. Any LAN hosts behind the connection, may even need to be lower −− 1452 or maybe even 1412.

- *Ping by IP address* works, but not hostname. The nameservers are not being setup correctly in `/etc/resolv.conf`. Check your client's (DHCP, PPPoX) documentation or enter these manually with a text editor. Get the correct DNS server addresses from your ISP.
- *PPPoX disconnects*. Unfortunately, there is a tendency for PPPoX to drop connections. PPP can be sensitive to any line condition which results in a temporary interruption of the connection. This may not be completely solvable, depending on what and where the problem is. You might try a different client, or check your current client's documentation on this issue. If worse comes to worse, set up a cron job to watch the connection, and re−establish if necessary.
- *Interface or route goes down for no reason*. If **ifconfig** and/or **route** show the interface and/or route has automagically disappeared, it may be due to a buggy NIC driver. This may also happen with DHCP if the server does not respond for long periods of time. (Possibly a bug in the client? I have seen this with early versions of **pump** from Redhat. HB.)
- Sub−par performance, or errors with the interface (e.g. eth0), may possibly be caused by a duplex mismatch. This would be most apparent when maxing out the connection. Most DSL modems and routers typically are set to half duplex, and your NIC should be set likewise.

5.5. Measuring Throughput

One of the first things most of us do is check our speeds to make sure we aren't getting short changed, and that our system is up to snuff. Doing this accurately is easier said than done however. First, remember you are losing 10−20% right off the top due to networking protocol overhead. Just how much is "lost" here depends on your provider's network architecture, where and how you are measuring this and other considerations. Most of us may wind up being closer to 20% than 10%.

Then, any time you hit the Internet, there is some slight degradation of performance with each hop you take. Now this may not amount to much, as long as you are not taking too many hops and all the components −− your system, your ISP's network, your ISP's upstream provider, and the destination itself −− are all working like well oiled machines. But there's the rub −− how do you really know with so many variables in the mix? One flaky interface, on one router, on one hop along the path, may cause misleading results.

Your absolute max speed is going to be at your point of connection to your ISP −− the ISP's gateway. It can only go downhill from there, not up! So the ideal test is as close to home as possible. This eliminates as many unknown variables as possible. If your ISP has a local ftp server, this is an excellent place to run your own tests. (Run a traceroute though just to see how local it really is.)

If your ISP does not have this, look for an ftp site that is close −− the fewer the hops, the better. And look for one that isn't too busy, or you will get misleading results. Find a large file −− like 10 Megs −− and time the download. Try this over several days, and at different times of day. The server, and the backbone, are going to be busier at certain times of day, which can skew results and you want to eliminate these variables as much as possible. Your provider cannot compensate for heavy backbone traffic, backbone bottlenecks, slow or busy servers, etc.

There are many test sites scattered around the web. Some are better than others, but take these with a grain of salt. There are just too many variables for these tests to reliably give you an accurate snapshot of your connection and throughput. They may give you a general picture of whether you are in the ballpark of where you think you should be or not. One good speed test is http://www.dslreports.com/stest/0. Another test is http://speedtest.mybc.com/ (both are Java). I find these to be better than some of the others out there.

Now keeping in mind that we are limited by the ~10−20% networking overhead rule, here is an example. My speed is capped at 1472 Kbps sync rate. Minus the ~15% is 1275 Kbps. My sync rate is known to be good and my distance to the CO is about 11,000 Ft, which is close enough that I should be able to hit my real world maximum throughput of 1275 Kbps or roughly 1.2−1.3 Mbps −− all other things being equal. From dslreports.com speed test:

```
Test running..Downloaded 60900bytes in 5918ms
Downloaded 696000bytes in 4914ms
First guess is 1133kbps
fairly fast line - now test 2mb
Downloaded 1679100bytes in 11090ms
Upload got ok 1 bytes uploaded
Uploaded 1bytes in 211ms
Upload got ok 1 bytes uploaded
Uploaded 1bytes in 205ms
Upload got ok 1 bytes uploaded
Uploaded 1bytes in 207ms
Upload got ok 50000 bytes uploaded
Uploaded 50000bytes in 2065ms
Upload got ok 100000 bytes uploaded
Uploaded 100000bytes in 3911ms

** Speed 1211(down)/215(up) kbps **
(At least 24 times faster than a 56k modem)
Finish.
```

1.211 Mbps is probably about as good as I can realistically expect based on my service. There is no reason for me to go troubleshooting or looking for tweaks.

Big Caution: my ISP uses a caching proxy server for web pages. This is a big equalizer for these kinds of web based tests. Without that, I surely would have been significantly slower on this test. The effect of the proxy is that you are actually testing throughput from the proxy −− NOT the test site. Just FYI. Another note: at the same time I tried another test site and was consistently getting 600−700 Kbps. So YMMV with these tests. (Usually I get the same on each, more or less.) Timing a large ftp download from two different sites, I calculated about 1.25 Mbps.

6. Appendix: DSL Overview

DSL is a telephone loop technology that uses existing copper phones lines, and provides a dedicated, high speed Internet connection. One of the big advantages of some DSLs (notably ADSL), are that they can co−exist on the same line with a traditional voice service or "POTS" (Plain Old Telephone Service). This is accomplished by utilizing different frequency ranges above the voice range (voice is up to 4KHz). Essentially, this gives two lines in one: one for POTS, and one for Internet connectivity. When all is working normally, there should be no interference between the two "lines". This gives DSL a potentially broad consumer base, and helps minimize costs for providers.

DSL is positioned for the Home and Small Office (SOHO) market that is looking for high speed Internet access at reasonable prices. Since it also typically provides dedicated, "always on" access, it can be used for interconnecting low to mid range bandwidth servers, and provides a great access solution for small LANs. It is also great for those Linux power users that just want a fat pipe :−).

Phone companies, and other independent telecommunications providers (CLECs), are now deploying DSL as fast as they can to stay ahead of the Cable companies −− the main consumer and SOHO competition for DSL providers. This mad rush to get "a piece of the pie", is bringing much competition (a good thing!), much diversity, and some confusion, into the consumer market. The DSL provider (often, but not always, the phone company) will provide the DSL infrastructure. This would include your line, the DSLAM, and physical connection to the outside world. From there it is typically picked up by an ISP, who provides the traditional Internet services.

Consumer DSL plans are typically "best effort" services. While boasting speeds approaching T1, and even surpassing that in some cases, it is not necessarily as reliable as T1 however. Business class DSL offers more reliability at a higher cost than consumer plans, and is a good compromise where both reliability and bandwidth are at a premium. All in all, the cost of DSL compared to traditional telco services, such as T1, is attractive and substantially more affordable for home and small business users.

DSL providers often do not have service contracts for home users, while business class DSL services typically do include similar SLAs (Service Level Agreements) to that offered for a T1 line.

The downside is that DSL is not available everywhere. Availability, and available bit rate (speed), are purely a function of where you live, where the telco has installed the prerequisite hardware, how far you are from the DSLAM/CO, and the quality of your phone line (loop). Not all loops are created equal, unfortunately. The primary limitation is distance.

6.1. The DSL Family

- **ADSL**

  Asymmetric Digital Subscriber Loop currently supports downstream rates up to 8 Mbps, and upstream of 1024 Kbps, hence the "asymmetric". The most widely deployed form of DSL at this time, and was specifically developed for the home and SOHO markets. The higher downstream rates lends itself to those not running serious servers −− at least anything more than a small, personal web site. ADSL is capable of sharing data with a POTS voice line, so an additional line is not required. A big selling point. ADSL, like other DSLs, is limited by distance. 18,000 ft (5.5 km) is a typical cut−off point for telcos. ADSL does typically require either a splitter or filters to isolate the DSL signal from POTS. Sometimes referred to as "full rate" ADSL in order to differentiate it from G.Lite DSL. There are two common line encodings for ADSL: DMT and CAP. DMT (a.k.a. Alcatel compatible) has won the standards battle and is now the standard and the more common of the two. Also, note that modems must be compatible with the encoding. In other words, a CAP modem will not work with a DMT service, and vice versa.

- **G.Lite**

  G.Lite is sometimes referred to as "DSL Lite", "Universal DSL" or "splitterless ADSL", is a slower version of ADSL that requires no splitters *or* filters. The isolation of voice and data signals is handled at the CO. Currently G.Lite supports speeds up to 1.5 Mbps/512 Kbps, and at one time was expected to become the dominant consumer DSL service. As of this writing, it is not nearly as wide spread as "full rate" ADSL however.

- **SDSL**

  Single−pair Digital Subscriber Loop, or also sometimes referred to as "Symmetric Digital Subscriber Loop" since it is indeed symmetric with a current maximum rate of 1.5 Mbps/1.5 Mbps. SDSL requires a dedicated line, and thus true SDSL is not as readily adaptable to the consumer market as ADSL. SDSL also uses a 2B1Q encoding (same as ISDN and some T1) which is considered more robust than the DMT or CAP encoding of ADSL. True SDSL is generally considered more of a server quality DSL. It is worth noting that some providers may be marketing a "SDSL" service that is really ADSL pinched so that upstream/downstream are the same. Or vice versa, SDSL with asymmetrically allocated bandwidth. Wasn't all this confusing enough already?

- **IDSL**

  ISDN Digital Subscriber Loop, 144 Kbps/144 Kbps is really a new and improved ISDN from Lucent Technologies and uses the same 2B1Q line encoding as ISDN, SDSL and others. IDSL does require a dedicated line however. The benefits are that it is an "always on" technology, like other DSLs, and provides an additional 16 Kbps over traditional ISDN. It is being marketed by some DSL providers as a low end bit rate option, where line quality is not sufficient for higher speeds such as that of ADSL.

- **RADSL**

  Rate Adaptive Digital Subscriber Loop was developed by Westell and has a potential of 2.2 Mbps downstream and 1.0 Mbps upstream. What makes RADSL more flexible is that the sync rate can be dynamically adjusted up or down as line conditions change. This makes it more of a viable alternative where line conditions are marginal due to distance or other factors. In many respects, RADSL is an enhanced ADSL to meet a more diverse set of line conditions. Like ADSL, RADSL can piggyback on the POTS line. RADSL does not require any splitters or filters.

-

HDSL

High bit−rate DSL was one of earliest versions of DSL. HDSL requires multiple, dedicated wire pairs, and is symmetric at 1.5 Mbps/1.5 Mbps (the speed actually depends on number of wire pairs used). Not a viable alternative for the consumer or SOHO markets.

- **VDSL**

  Very high rate Digital Subscriber Loop is a DSL still in development with a current downstream capacity of 52.8 Mbps, and upstream of 2.3 Mbps. At this time, VDSL is limited to very short loop lengths, and is not yet a viable alternative. It may find application where there is fiber to the neighborhood, and thus the copper loop segment is relatively short.

- **UDSL**

  Unidirectional Digital Subscriber Loop is a proposal from Europe that is not yet in use.

- **G.SHDSL**

  The standards for G.SHDSL are not finalized yet, but should be soon. Supposedly to include many enhancements including better reach and better rate adaptation.

6.2. The DSLAM

This technology is made possible by the placement of DSLAMs, or Digital Subscriber Loop Access Multiplexers, from such suppliers as Alcatel and Cisco, in the telco's Central Office, or sometimes a suitable remote location. DSLAMs come in various shapes and sizes, and are the one, single complex and costly component of a DSL connection. When a qualified phone line is connected to a modem at the user's end of the loop, a high speed digital connection is established, typically over ATM, or sometimes frame relay. The DSLAM splits the signal back into separate voice and data channels. The voice channel stays within the telco network, whereas the data is picked up by an ISP (typically).

Figure 4: A Typical DSL Connection Path

```
Voice -+                                               +---> Voice
       |<-- copper loop --> DSLAM/CO <--{ATM cloud}--->|
modem -+                       |                       +---> Inet
       |                       |                       |
ether..|..... DSL/ATM here ....|.... raw ATM here .....|.. TCP/IP ..
       |                                               |
SOHO...|............ telco (ILEC or CLEC) .............|.. ISP ..| NSP
```

6.2.1. Sync

A good, working connection to the DSLAM is referred to as "syncing". This is typically indicated by LEDs on the modem. Without sync, nothing happens. The modem will establish a sync rate which is often throttled by the provider at a predefined limit. This limit, or "cap", is at the provider's discretion and is part of the service that is being provided. Your modem may well sync at a higher rate than the "cap", but your speed will be limited to whatever "cap" the provider is enforcing. So while ADSL has an upward theoretical limit of 8 Mbps, you will not see that speed −− unless of course your provider is selling an 8 Mbps plan. Most plans are well below this.

Below is the status information from a SpeedStream 5660 modem/router via the built−in telnet interface. In this example, the customer is on a 1.5 Mbps/384 Kbps service:

```
Command−> show dslstatus

−−− Channel Info               ATU−R                    ATU−C
 Current TX Rate  −           384000                  1500000
 Previous TX Rate −                0                        0
 CRC Block Length −                −                        −
 Interleave Delay −                −                        −

−−− Physical Layer Info        ATU−R                    ATU−C
 Current Attainable Rate −    448433                  3890243
 Current SNR Margin      −      10.5                     17.0
 Current Attenuation     −      54.5                     31.5
 Current Output Power    −       3.0                     16.0
 Current Status:
  Defects detected       −        No                       No
  Loss of Framing        −   No Loss                  No Loss
  Loss of Signal         −   No Loss                  No Loss
  Loss of Power          −   No Loss                  No Loss
  Loss of Signal Quality −   No Loss                  No Loss

−−− ATU−R Line Status
 Line Coding − DMT
 Line Type   − Fast or Interleaved

Command−>
```

First notice the "Current Attainable Rate" in the "ATU−C" column. This is the downstream sync rate negotiated by the modem and DSLAM, which is over 3.5 Mbps. The actual speed is limited, however, to 1.5 Mbps/384 Kbps from the first row "TX Rate". This is the theoretical limit of this connection. This limit, or "cap", can be enforced at the DSLAM, as is the case the here, or further upstream. Had the first row "TX Rate" been lower than the provider's imposed limit, then this would indicate some kind of problem with the connection, perhaps due to distance or some kind of line impairment.

The attainable sync rate is the result of a number of factors, including wire distance to the DSLAM, quality of both inside and outside wiring, the loop wire gauge and various other factors within the loop. Actual measurable, real world throughput, on the other hand, is first of all dependent on sync rate. Low sync rate means low throughput. In the above example, had the sync rate been lower, say 500 Kbps, then that would be the maximum for that connection, even though the customer is paying for a 1.5 Mbps service.

Secondarily, throughput will depend also on the ISP's network, and then the ISP's upstream provider. You will lose approximately 10−20% of potential throughput to networking overhead. In the above example where the connection is throttled at 1.5 Mbps, the actual, real−world maximum throughput would be somewhere around 1.2−1.3 Mbps when overhead is taken into account. Moreover, once you hit the Internet proper, all bets are off as there are any number of factors that may impact throughput. A overloaded or busy server is likely to be slow no matter how fast your DSL connection is.

6.3. DSL Modems

The modem is the last piece of the connection. The modem is connected directly to the DSLAM via the copper loop on the telco end, and plugs into a wall jack on your end. When all is well, the modem "syncs" with the DSLAM, and then makes an IP connection to the ISP, and off we go!

For Linux users, *the modem is a very important consideration*! There are many modems supplied by ISPs that are not Linux compatible. Your best bet is an external, ethernet interfaced modem (or modem/router combo) that connects via a standard ethernet NIC, since most other modem options (PCI, USB, onboard) will not work due to a lack of drivers at this time! All ethernet based modems will work fine. (See the Modems Section for an up−to−date list of compatible modems.)

With ethernet modems, the only potential compatibility issue is the Network Card (NIC). (And really any compatible ethernet NIC should do just fine −− 100 Mbps is not even necessary.) You are probably better off anyway, since PCI and USB modems tend to be more problem prone. If your chosen provider does not offer a compatible modem as an option, then you either need to look elsewhere, or you will have to buy one outright from a third party.

As always, there are exceptions. Xpeed now has drivers for two PCI modems included with the kernel drivers (as of 2.2.18). These are the first open source Linux DSL modem drivers, and is welcomed news. Alcatel's ADSL SpeedTouch USB modem now has Linux drivers. Diamond makes an internal PCI modem which has binary−only drivers, but it is not in widespread use. Efficient Networks has reportedly been working on Linux drivers for their SpeedStream 3060 and 3061 PCI modems. It is also possible to make a direct ATM connection using a modem plus an ATM card, though this delivery system is not used in the U.S. as far as I know, and should not be considered as a viable option. This would also require a 2.4 kernel.

The most common type of modem in use today is actually a combination "bridge" and modem device. The bridge is a simple device, typically with little configuration. Network traffic passes blindly across the ATM to ethernet bridge in either direction. Your point of exposure is the interface (typically a NIC) that is connected to the modem/bridge.

Some ISPs are also be offering "routers". These are basically combination modem/routers that can handle NAT, and may have other feature enhancements such as port forwarding, a built in hub, etc. These are all external, so should work too. But probably not a big deal for Linux users, since Linux can do anything these do, and more. A locked down Linux box makes a most excellent firewall/gateway/proxy!

To confuse things even more, there are also all−in−one devices: combo bridge+router+modem, sometimes called "brouters". In this case, the modem can be configured for either bridged or routed modes −− but it can't be both at the same time.

All providers should make available a modem of some sort. Many ISPs will have more than one modem option. Some may give away the modem at no additional charge. Some may offer a free base model, and charge the difference for the better models with more features. Many of the modems that ISPs supply are not available through normal retail channels. Should you want to buy one yourself, this leaves used equipment outlets (e.g. ebay), or possibly buying a modem that your ISP may not support (i.e. a possibility of no tech support if you have a problem).

While some ISPs provide modems that are not readily available through normal retail channels, there are a number of manufacturers that are getting on the DSL modem bandwagon, and offering a good selection. Most have a number of enhancements. At this time Alcatel, Intel, Zyxel, Cisco, 3Com, and Cayman have products available. Depending on model and feature set, prices range from a little over $100 US to $800 and up. Many of these handle their own authentication and encapsulation (DHCP, PPPoE, etc).

Are some modems better than others? Short answer: no. Modems are not much of a factor in speed. But some do have enhanced features, such as diagnostics or the combo modem/routers. Ethernet modems are generally considered the most reliable. Fewer IRQ hassles, no buggy drivers, etc. So the fact that Linux users are mostly relegated to ethernet modems is a blessing in disguise really. Are any of these better than others? Hard to say since most of this is so new there is not enough of a track record to compare brands and models with any degree of assurance. In other words, any old external, ethernet modem should do −− provided it matches your provider's DSL, and is configured for that service. Remember, there can be differences here.

Warning

Make sure any third party modem or router you may purchase is compatible with your DSL provider. There are two major line encodings for ADSL (CAP and DMT a.k.a. Alcatel compatible), and several options for IP encapsulation. And different DSLs (SDSL, IDSL, etc) will require their own modems too. Your provider should have a list of compatible options. It may well have to be configured for your ISP's service too. Don't expect it to work right out of the box either (unless it does indeed come from your provider). Many are accessible via telnet, or a web browser, where the configuration options are available. See the owner's manual for this.

6.4. The ISP Connection

The modem connects to the DSLAM, and then the DSLAM is connected to the telco's ATM network (or frame relay), where it is picked up by the ISP. The ISP typically will take over at what we "see" as the first hop on a **traceroute**. Everything up to that point is in the hands of the telco/DSL provider. The ISP will connect to the telco's ATM network via a high−speed data connection, usually ATM over a DS3 (45 Mbps) or possibly an OC−3 (155 Mbps). The important thing here is that an ISP must "subscribe" with your telco to provide this connection. The ISP will provide traditional ISP type services: email, DNS, news, etc. It is really a two step connection −− DSL from one provider, Internet from a second −− even though these may be combined into one billing.

The Baby Bells (RBOCs) in the U.S. all own ISPs. These, of course, are connected to their DSLAMs, and are providing Internet services via the telco's ISP subsidiary. Many independent ISPs are availing themselves of the ILEC's DSL services, and in essence "reselling" the DSL services of the ILEC. While the underlying infrastructure is the same in this case, having more than one ISP working out of a CO may mean a better selection of features and prices for the consumer.

CLECs (independent telcos) are now installing their own DSLAMs in many U.S. markets. This makes them a direct competitor to the ILEC. In this scenario, there would be two (or more) DSL providers in the same CO, each with their own DSLAM(s), and each competing against each other. This complicates the ISP situation even further, as each DSL provider will be "partnered" with one or more ISPs. If you are lucky here, you will have many choices of plans and pricing structures.

At this time, there is a shake out going on in the U.S. market. The independents are all struggling to match the deep pockets of the regional phone companies. The independents that have survived are now focusing more on small business and higher−end consumer customers. This means, it will cost more, but you should also expect to get more. Especially, in the quality department.

Typically, your service agreement is with the ISP, and not the DSL provider. This makes the actual DSL provider a "behind the scenes" player. This may vary, and in some cases, you may wind up with a separate service agreement for both the DSL provider and the ISP.

See the Appendix for a list of Linux Friendly ISPs.

6.5. Availability

Who can get DSL? The first requirement is that a telco has installed the necessary hardware somewhere on their end. Typically this is in the CO. You have no choice on which CO is yours −− it is wherever your loop terminates. If your CO has a DSLAM, and the necessary other components, then DSL may be available to you. This is often known as "pre−qualifying", and is Step One in getting service.

More and more "remote terminals (aka DSLAMs)" are being deployed. This is certainly good news for those that are a long way from the CO. CO distance is not the limiting factor it once was.

6.5.1. Ordering

Before ordering service, check to see what providers there are in your area. Look for options on both the telco/DSL side and the ISP side. You may have several options, including the large phone companies, as well as smaller, local ISPs. Once an order is placed, you must wait for the qualification process before a provider will agree to provide service.

6.5.2. Qualifying

Once local availability is established, the next step is "qualifying" your loop. The provider will run various tests to make sure that your loop can handle the DSL signal. This is to determine how suitable your line is for DSL, and maybe what level of service will be available to you. You probably will have to order service just to find out this much. It can be a fairly involved process, with a variety of different tests being run. There are a number of things that may "disqualify" a line. The most common limitation is distance.

All DSLs have distance limitations. ADSL is limited to a loop length of roughly 18,000 ft (5.5 km), but the actual cut off point will vary from provider to provider. The further away you are, the weaker the signal, and the potential for poor connections is greater. With ADSL, if you are within approximately 12,000 ft (3.7 km), you should be able to get at least 1.5 Mbps −− all other things being equal. IDSL has even greater reach, mainly because the maximum speed for IDSL is considerably lower at 144 Kbps/144 Kbps.

Still even if you're close enough, there are a number of potential impediments that may disqualify a line. Two such common impediments are load coils and bridge taps. These are aspects of the old telco infrastructure that once were deemed beneficial, but now are getting in the way of the newer, digital technologies. Whether you hit a snag like this or not, is pretty much hit or miss. Fiber anywhere in the loop is also a disqualifier. The provider may take steps to "clean" the line. Just how far they are willing to go will vary from provider to provider, and this will likely add additional time to the installation process.

Once the line is "qualified", the next step is deciding on which plan is suitable for your situation. The provider may have differing plans available depending on how strong a signal they think your line can handle. If you are marginal, you will not be qualified for the higher speed plans. And if price is a factor, having a tiered pricing structure is good also since the lower end plans are obviously less expensive. How this is structured also varies wildly from provider to provider. Since, DSL is a new service, and providers are trying to find the right price/feature combinations that will attract the most users and thus gain a competitive edge.

Some common data rates:

```
     Downstream/Upstream

       128 Kbps/128 Kbps

       256 Kbps/256 Kbps

       384 Kbps/128 Kbps

       640 Kbps/90  Kbps

       1.5 Mbps/384 Kbps

       2.0 Mbps/512 Kbps

       7.1 Mbps/1024 Kbps
```

and a near infinite number of other possibilities. The cost of different plans generally goes up with their speed.

Should you be disqualified, and have other options, get a second opinion. Calculating the effective loop length is by no means an exact science. There is plenty of room for errors. Also, some providers may go to greater lengths to "clean" the loop than others. And, if you have more than one phone line, and are disqualified, then try the other line. Just because they both terminate at your location, does not necessarily mean they are the same length! The telco network is full of surprises.

6.6. Choosing Providers

Should you have more than one choice, here are some things to keep in mind when comparing services from different providers. If you are in a populous area, chances are you do have a number of choices. There is a dizzying array of possibilities at this time. Remember too, that it is a two step connection: DSL provider and ISP. You may have choices for each.

- *A compatible modem*. For now with Linux (or any alternative OS) this essentially means an ethernet interface. (There are rare exceptions.) "Routers" (i.e. combo modem/routers) should be OK too since these seem to be all ethernet devices.
- *Installation*. A self−install option, of course, let's anyone get up and running, and is less expensive. But if there is no self−install available, will the the provider install onto a Linux only site? Many will not! Having a Windows (or Mac) box temporarily available is a work around here. Even a laptop may be enough.
- *Static vs Dynamic IP Address*. If wanting to run servers, or hosting your own domain, static is the way to go. A dynamic IP, on the other hand, makes you a little harder to find should you wish to remain "invisible", or a least harder to keep track of.
- *Encapsulation*. Is the connection "Bridged" or "PPP". PPPoX has the reputation of being not as stable a connection, and not "always on". PPPoE requires client software to manage the connection, so one more layer of code.
- *Server Policy*. Some ISPs are fairly open about this, while others forbid any servers −− even personal web sites. Others may even go so far as to block certain ports.
- *Contract*. Is there a contract, and what are the out clauses? Cancellation fees?
- *Connection Limits*. Is it "always on" (at least theoretically :−)? Are there session limits, or idle timeouts? Is bandwidth metered and limited to so much per month? Do they forbid a LAN behind the connection (dumb!)?
- *Linux Support*. A few ISPs may offer some degree of tech support for Linux, but most will not. This isn't so bad, as long as they don't go overboard and refuse to help with anything just because you run a non−supported OS. ("Supported" means like "tech support".) If they say "we don't care", you should be good to go.
- *Free Dialup Account*. A nice thing to have if the connection is down, or you just need to check mail from another location.
- *Setup program*. A few ISPs may have a setup program you are required to run the first time you connect in order to setup your account. This will likely not have a Linux version. (BellAtlantic.net was doing this at last report.) Other than this, there is nothing proprietary about DSL, and related protocols.
- *Reliability and Quality of Service*. Ask around in your local area from those that have the same DSL provider and ISP. A local LUG is a good place to get this kind of info. How much down time (hopefully not much)? Are mail and news services good? Backbone routing? Tech support?

There are a number of other options and features that might be worth looking at too: multiple IPs, domain hosting (DNS), free web space, number of email accounts, web mail, etc. All things considered, the better plans are probably going to cost more for a reason.

7. Appendix: FAQ

Some Frequently Asked Questions about DSL and Linux.

1. Q. Does DSL work with Linux?

   DSL is a technology, or more correctly, a group of related technologies. This is akin to asking if Linux works with telephones. The technology itself does not care. So, the short answer is "Yes, of course!". The long answer is that if there are any impediments, they are being imposed by the provider. There are things they may do, that can make getting Linux up and running, more of a challenge than it needs to be. Not having a compatible modem option available is one common gotcha. Also, if the telco or ISP is doing the installation, they may require a Windows or Mac system to be available. This saves them the costs of training their techs on various alternative OSes. Buyer beware!

   Basically all DSL does, is facilitate a high speed Internet connection. At some point, it is all TCP/IP, and Linux, of course, handles TCP/IP quite well.

2. Q. Where can I find drivers for my PCI (or USB) modem?

   With few exceptions, you probably can't, because they are just not available. Your best bet is an external, ethernet interfaced modem for all intents and purposes. If your provider does not offer one, you will have to find another provider, or buy your own modem outright. Just make sure it is compatible with your provider's flavor of DSL.

   The are exceptions to every rule. See the Modems Section for a list of compatible modems as of this writing.

   If an incompatible modem puts you in a bind, hopefully you will take the time to politely harass the manufacturer ;−).

   This situation may be changing. Xpeed now has drivers included in the kernel for source for their PCI IDSL and SDSL modems. This is good news! Alcatel has just recently released drivers for the Alcatel SpeedTouch USB ADSL modem. Hopefully others will follow suit. (Make sure you are reading the latest version of this document, as I have intentions of keeping this situation updated as needed.)

3. Q. How fast or good of a network card do I need?

   Any card that is compatible with Linux should work fine. Remember even low−end cards are 10 Mbps, and no consumer class DSL is near that at this time. I would suggest a reasonably good quality card, just to help eliminate the possibility of errors and premature failure.

4. Q. How can I find out when DSL will be available in my area?

   Just where and when DSL gets deployed is totally in the hands of your friendly local telco. They obviously can't do everyone at once, so they probably are selecting areas based on competitive factors. Getting a straight answer from a telco on this question can also be a challenge. Probably so as not to tip their hand to competitors. Unfortunately, it is a question only they can answer.

5. Q. I was disqualified because I am too far away. What can I do?

   Move? Seriously, there isn't much you can do. If there are other providers, get another opinion. You never know. Determining the loop length is an inexact science, and there is room for errors. Many use databases for this, and these databases routinely have some inaccuracies. Some providers too, may be more aggressive in taking steps to help you out and clean up the line. Also, some providers offer low−end speed services that have greater reach. Maybe this will become available in your area. Or, the telco may install, at some point, remote devices for customers who are now too far away.
6. Q. What are the speed tweaks for Linux?

   This may not be necessary. Linux is pre−tweaked for the most part, unlike some versions of Windows that really need some registry hacks to get optimum performance. If you have a high latency connection, you may benefit from increasing the TCP Receive Window. See the Tuning section.

   Now if you are convinced you are not getting the performance you should based on your distance and line conditions, then maybe there is a problem somewhere. See the Troubleshooting section for more. What you need is a fix, more than a tweak.
7. Q. My service is limited to 640K (for example). Can I get better speed by getting a faster modem? Any way around this?

   No, and no. The modem has little bearing on how fast your connection is for all intents and purposes. The provider has a mechanism in place for limiting your speed somewhere in the pipe before you hit the Internet. There is no way to defeat this.
8. Q. Can I download and upload at the same time? Is one effected by the other?

   The upstream and downstream channels use separate frequency ranges within the DSL signal, so simultaneous upload/download is not a problem and available bandwidth is not normally impacted.

   Where there may be somewhat of an adverse impact, is with asymmetric DSLs like ADSL, *and* both the upstream and downstream are simultaneously saturated. This is a TCP 'feature' and not DSL related though. This can adversely effect the faster stream (i.e. the downstream). How much of an impact depends on a slew of factors and is beyond the scope of this document, but is more pronounced with higher ratios of downstream to upstream (e.g. 640/90).
9. Q. I am paying for 768 Kbps service, and the best I ever get is 640 Kbps or so. Why? Is the service oversold? I am not getting what I pay for.

   You will lose 10−20% of the rated capacity due to the overhead inherent in the various protocols utilized. Most of us will probably fall closer to 20%. This is just a fact of life for everybody. Just how much is lost here depends on a number of factors, and may vary from provider to provider. You seem to be close to your maximum when this is taken into consideration. Also, if you read the fine print, many ISPs are advertising speeds "up to" such and such. Check your service agreement and see if there are any guarantees. If there are, they may be well below the advertised maximum speed, and may be based on sync rate instead of actual throughput. Though this may vary from provider to provider as well.

Also, be careful how you test this. Some of the so−called test sites can be pretty unreliable. There can be many factors between you and that site that can impact your throughput and skew results −− not the least of which is how many people might be trying that same test at the same time. The best test is via FTP download from a known good, close, not too busy site.

10. Q. Why does PPPoX have such a bad rap?

    The occasional disconnects is one of the biggest gripes. PPP seems to be sensitive to any interruptions in the connection. Generally a disconnect means a new IP. And there are those that say PPP, by its very nature, was never meant to be an "always on" protocol. PPP is a session management protocol at heart, that requires a user to initiate a connection and authenticate him or herself. PPPoE/A are not yet particularly mature protocols either. They do not have much of a history or track record. Some would say the telcos and hardware manufacturers have rushed this out the door. PPPoE also requires an additional layer of software just to maintain the connection. This is one more layer of code and one more potential point of failure. Also, more system overhead is utilized to manage the connection.

11. Q. Why PPPoX? This seems like a bad idea!

    PPP gives several advantages to the provider: they can use their existing infrastructure and hardware that they now use for their (larger) dialup customer base. It is easier to control user authentication and potential abuse situations, and easier to manage their network and related issues. In fact, it most boils down to its just easier for them. Easier, means saves man hours, and therefore saves costs (at least from their perspective).

    It is not a conspiracy to conserve IP addresses, or thwart heavy users. IP address costs are insignificant in the overall scheme of things.

12. Q. The only provider in my area does not support Linux. What can I do? Will I have to use Windows?

    NO! "Support" here is support as in "tech support". They are just saying that they will not give you tech support when and if you have problems. This does not mean you cannot use Linux on their network. Just that you may have to fend for yourself when and if a problem does arise. Anything that is forbidden will be in their Acceptable Use Policy (AUP), or Terms of Service (TOS) agreement.

    I have heard stories where a new tech or installer has misinterpreted their own company's policy on this and told someone "you can't use Linux here". Same with NT server. But this is almost always a misinformed individual.

    But −− if a provider does not support Linux, they may balk at installing onto a Linux box. Hopefully, they will have a self−install option to get around this annoyance. YMMV.

13. Q. My fax software does not work with my DSL modem. Why is that?

    Faxes are normally transmitted over typical analog phone lines by dialing the fax machine on the other end. Analog modems can handle this, but DSL "modems" have no dialing capability. Don't throw out that 56K yet!

14. Q. What does "FastPath" mean? Is it better? Faster? What is interleaving? How can I get better ping times?

Interleaving is a feature of DMT line encoding. Essentially it is a form of error correction that is configurable at the DSLAM. The side effects are a slower connection, especially higher latency. With FastPath (or sometimes called non−interleaved) DMT, gateway pings can be in the 10−25 ms range. With interleaving, this is more likely to be in the 40−75 ms range depending on the degree of interleaving that has been enabled.

On the positive side, a marginal line is more stable and less prone to errors with interleaving. Many telcos have interleaving on by default since increased stability would seem to be a good thing. But this is only beneficial for marginal lines, and everyone else is paying a latency tax for this. Some telcos may be amenable to turning this feature on/off. YMMV.

15. Q. How fast and powerful of a computer do I need for DSL? My ISP says I need at least a Pentium 200. Why?

    At the most basic level, a 386 will work fine. In most situations, you are connected to what is essentially an ethernet based network. So theoretically anything that can handle a very slow ethernet connection would work. No comment on well Netscape will run on a 386 though ;−) But as far as just managing a raw connection, a 386 is indeed workable. What else you can do with it, is another matter.

    Where this gets a little more complicated is the modem, and the client that the ISP may require. Any PCI or USB modem is going to require drivers, which means more CPU and system resources. Also, PPPoE does even more processing, so again the potential CPU load is increased. Windows tends to be not so efficient with all this going on, hence the requirement for mid range Pentiums by some ISPs.

    With Linux it will depend on what you are going to do. A low end Pentium should be fine for most uses. A 386/486 should do nicely for just a firewall/gateway box. Just remember if you are running PPPoE, you may take a performance hit on low end hardware.

16. Q. I just got my DSL installed, and my speed sucks, and/or my connection constantly drops. What is the problem?

    Not enough information to say, really. There are many, many things that can cause a poor connection. The list is too long to mention them all.

    One of DSL's weaknesses is that the signal can be fairly fragile. Many things can degrade the signal, making for poor connections, and thus speed. This can be caused by poor or substandard inside wiring, a wiring problem outside (like bad splice), RFI from any number of sources, AM radio signal interference from a nearby station, bridge taps on your line, excessive distance from the DSLAM and so on. Not to mention possible hardware problems with your modem, NIC, or the telco's DSLAM, etc. Not always easy to sort out.

    Your provider should be able to assist you. First, make sure the problem isn't with your setup as they likely won't help solve a Linux problem. Then be persistent, and don't hesitate to go over someone's head if the help is not forthcoming. Most problems are solvable. The trick is isolating it. A good telco tech, trained for DSL, can find all kinds of obscure wiring problems.

17. Q. My provider's tech support staff is clueless. What can I do?

Common complaint. Seems to be the nature of the beast. First line tech support is an entry level position, and mostly filled by young people with little technical or networking knowledge. Grin and bear it, or try calling back.

18. Q: Are there ADSL Standards?

    Sort of. The U.S. Bell Operating Companies have standardized on Discrete Multi−Tone (DMT) (ANSI T1.413) in their current roll−out. Most others should follow their lead in the states. There are other types of modems, most notably Carrier−less Amplitude Phase Modulation (CAP), which of course, is incompatible with DMT.

    A biased comparison from an DMT−based vendor on this subject can be found at the http://www.aware.com. Still, it provides the best detail on this issue I have seen so far.

    A rather expensive copy of the ANSI standard can be ordered at: American National Standards Institute ANSI Home Page

    Asymmetric Digital Subscriber Loop (ADSL) Metallic Interface

    ANSI TI.413−1995

    Note: ANSI TI.413 Issue 2 was released September 26, 1997

19. Q: Can I use ATM to connect to DSL?

    Technically speaking, you can. Some DSL modems (at least the Alcatel version) has a ATM Forum 25Mbps interface, which connects to a PCI ATM card. But this is rarely done in practice. See http://lrcwww.epfl.ch/linux−atm/ for more details.

20. Q: Why does DSL have all these bit rates (384/1.5/7.1M/20M/etc) options?

    The basic problem is the 100 year old design of the copper loop. It works great for analog phone, but it presents a real challenge for a digital signal. Remember that the distance of a loop is inversely proportional to the data rate that it can carry. Rate adaptive technologies are great for making a digital signal work in many situations, but it can't provide a consistent bandwidth for all applications, especially for very long (over 18,000 ft) loops. The different bandwidths that you see advertised reflect various marketing wars of vendors equipment, and the telco struggle to finalize on a standard set of data rates. The bottom line is for the telco to be able to reach as broad a customer base as possible.

    Check out the next question on the loop impairments that cause this to happen.

21. Q: What are all these loop impairments (bridge taps, load coils, DLCs) that could disqualify my line from DSL? (thanks to Bruce Ediger)

    Load coils: in−line inductances that improve voice−frequency transmission characteristics of a telephone circuit. Essentially, a "load" steals energy from high frequencies and gives it to lower frequencies. Typically only used in very long (> 9,000 ft) phone lines.

    By "bridges" I assume you mean "bridged taps". In older neighborhoods, the phone wiring will have been used by more than one customer. Perhaps these customers lived at different (though near−by) addresses. The unconnected "spur" of wiring is a "bridged tab" on the currently connected circuit.

DLCs, Digital Loop Carriers: there's a bunch of systems for carrying more than one voice transmission on a single pair of wires. You can shift the frequencies up or down, or you can digitize the voice transmissions and divide the telephone circuit by time or code or something. The more general term is "pair gain".

These things cause different problems for high−frequency communications.

Load coils will completely mess up things by filtering high frequencies and passing low frequencies. They probably also change the "delay envelope", allowing some frequencies to arrive before others. One byte's tones will interfere with the next byte's.

Bridged taps act as shunt capacitances if they're long in relation to the signals wavelength, and they'll actually act as band pass filters if they're about 1/4 wavelength of the signal. That is, they'll pass particular frequencies freely. Particular tones of a DMT modem might get shunted back, rather than passed along to the receiving modem, reducing bandwidth for that telephone line.

Pair gain, digital or analog, limit the bandwidth available to one transmission in order to multiplex several on one wire. High and low tones of a DMT transmission get filtered out by the apparatus.

The book "Subscriber Loop Signaling and Transmission Handbook", by Whitham D. Reeve, , IEEE Press 1992, ISBN 0−87942−274−2 covers the math of how to calculate the effect of line length, bridged tap, etc on the transmission characteristics of a telephone line. It's pretty expensive, however.

22. Q. Can I run a web server with my DSL connection?

    Sure. You are connected to a TCP/IP network, so theoretically you can run any service that the protocols allow −− mail, ftp, ssh, irc, etc. Where there may be problems, is with the ISP's TOS (Terms of Service). Some ISPs are pretty open on this, while others forbid any type of server, and may even block certain ports. You should research this, or ask the ISP before making any plans. ISPs that are selling a consumer service are not going to allow any high volume servers −− just personal, or low traffic services at best. If this does not fit the bill, then you can check with any local Business class DSL providers. This will cost more, but the Terms of Service, and guarantees, are generally much more suited to higher bandwidth usages.

    If you do not have a static IP, you can get around this with one of the many Dynamic DNS services that are out there for just this purpose. See the links section.

23. Q: Do you have examples of DSL Modems?

    Short Answer: Yes. Real Answer: The evolution of this technology is moving too rapidly for anyone to keep up to date in a HOWTO. Check http://dslreports.com/information/equiprated/all for up to date information.

    However, below is a list of some of the current modem offerings as of March 2001. All are ADSL modems with DMT encoding (a.k.a. Alcatel compatible), unless specified otherwise. [Note: Some items retained from original list dated June 1998.]

    ♦ Router/Modems with 10/100baseT Ethernet Interface:

Examples: Flowpoint 2000 DSL(CAP), 3COM Viper−DSL (CAP), Westell ATU−R−Flexcap (CAP), Aware x200, Zyxel P641, Efficient Networks SpeedStream 5660, Cayman 3220H, Cisco 673 (SDSL), Cisco 675 (ADSL/CAP), Cisco 677 (ADSL/DMT), Alcatel SpeedTouch Pro

- Bridge/Modems with 10/100baseT Ethernet Interface:

  Examples: Alcatel 1000, Alcatel SpeedTouch Home [note: there are also USB and PCI SpeedTouch versions!], Westell ATU−R−Flexcap2 (CAP), Efficient Networks SpeedStream 5260, Efficient Networks SpeedStream 5251 (SDSL), Westell WireSpeed.
- Modems with ATMF Interface:

  Examples: Alcatel 1000, Alcatel SpeedTouch Home, Cisco 677 (DMT), Ariel Horizon II
- Bridge/Modems with V.35 Serial Interface (T1, Serial Router)

  Examples: Westell ATU−R
- Modems with USB Interface:

  Efficient Networks SpeedStream 4060, Intel 3100, Alcatel SpeedTouch USB
- PCI Modems:

  Examples: Cisco 605, Efficient Networks SpeedStream 3060/3061, Intel 2100, Xpeed X200 (IDSL), Xpeed X300 (SDSL), Alcatel SpeedTouch PCI
- Wireless Modems (IEEE 802.11b):

  Examples: Alcatel SpeedTouch Wireless
- Dedicated Router (no built in modem) with 10/100baseT Ethernet Interface:

  Examples: Netgear RT311, SMC 7004BR, Linksys BEFSR11

This is but a very small sampling and should not be construed as endorsements of the products listed. It is just a simple illustration of a few of the available products.

Warning
Modem manufacturers often ship modems to meet an ISP's specifications. Features are sometimes enabled or disabled as requested by the ISP. There are conceivably numerous, possible variations on each model. Something to consider if buying one second−hand.

8. Appendix: Miscellaneous

8.1. Links

- Other related documentation from the Linux Documentation Project:
    - ♦ Firewall HOWTO
    - ♦ Security HOWTO
    - ♦ IPCHAINS HOWTO
    - ♦ IP Masquerade HOWTO
    - ♦ Home Network mini HOWTO
    - ♦ Ethernet HOWTO
    - ♦ Networking Overview HOWTO
    - ♦ Net HOWTO, previously named the NET3−4−HOWTO, the definitive, in−depth guide to various Linux networking topics.
    - ♦ Linux 2.4 Advanced Routing HOWTO. All the new, improved features are explained here.
    - ♦ DHCP HOWTO
    - ♦ VPN HOWTO
    - ♦ VPN Masquerading HOWTO
- More on the 2.4 kernel packet filtering from The Netfilter Project at http://netfilter.samba.org/. Several good HOWTOs for the new features in 2.4.
- Check your security and see what ports are open at http://hackerwhacker.com/. This is one of the better sites for this. Some only test a relatively few ports.
- SuSE's Linux PPPoE page is at http://www.suse.de/~bk/PPPoE−project.html. Good information on most of the available Linux PPPoE implementations.
- Bob Carrick's definitive PPPoE site is at http://www.carricksolutions.com/. His Linux PPPoE page is at http://www.carricksolutions.com/linuxpppoe.htm. It has some other DSL related information as well. All OSes are covered.
- The NTS EnterNet for Linux FAQ can be found at http://http://support.efficient.com/KB/NTS/linux.html. This is a non−GPL'd PPPoE client that is distributed by some ISPs.
- ATM on Linux: http://lrcwww.epfl.ch/linux−atm/. Where to find the latest info on PPPoA and raw ATM connections.
- A step by step report on getting Linux going with raw ATM is here: http://linux.com.sg/news/atm/.
- FreeSwan, http://www.freeswan.org, is an IPSec and IKE VPN implementation for Linux.
- VPN and Masquerading on Linux: http://www.wolfenet.com/~jhardin/ip_masq_vpn.html
- PPTP−linux allows you to connect to a PPTP server with Linux. The home page is http://cag.lcs.mit.edu/~cananian/Projects/PPTP/.
- Justin Beech's http://dslreports.com, a great site for anything and everything related to DSL. If it's not there, then there is a link to it. (Site runs on Linux.)
- John Navas's Cable and DSL site, http://cable−dsl.home.att.net, has good general info, tweaks, troubleshooting, hardware info, etc. for all OSes.
- TCP Performance Tuning tips: http://www.psc.edu/networking/perf_tune.html. Tips on Linux, and other OSes.
- A great Linux security site is http://linux−firewall−tools.com/linux/. Lots of info from Robert L. Ziegler, author of *Linux Firewalls*. Many links to other security related sites as well.
- http://www.securityportal.com/lasg/, The Linux Administrator's Security Guide by Kurt Seifried. Good tutorials on a variety of topics −− not just firewalls, but the big picture.
- The Seattle firewall is an ipchains based firewall that can be used on a dedicated masquerading

firewall machine (including LRP), a multi−function masquerade gateway/server or on a stand−alone Linux system. The project is located at http://seawall.sourceforge.net/

- My ipchains script is at http://personal.bellsouth.net/~hburgiss/linux/ipchains.html. This has IP Masquerading already set up, is reasonably well commented, and may make a quick starting point for your own script with only minor adjustments to suit your situation.
- Here a few pages dedicated to using Linux with specific providers. (I could use some submissions for more please.)
  - Verizon: http://www.panix.com/~dfoster/prog/linux/pppoe.html
  - Southwestern Bell: http://home.swbell.net/sdboyd56/DSL/connect1.html
  - BellSouth: http://personal.bellsouth.net/~hburgiss/dsl/survival/linux.htm
  - HomeChoice (UK): http://www.maxuk.net/hc/faq.html. (This gets my vote for the strangest ADSL service anywhere.)
  - France Télécom's Netissimo: http://www.rhapsodyk.net/adsl/HOWTO/. Good information on setting up PPTP with Linux for Alcatel modems.
  - Austrian Highspeed Internetconnection & Linux HOWTO: http://www.members.aon.at/heimo.schoen/at−highspeed−howto.html.
- Now that you have a full−time connection, want a routable hostname for your computer? Dynamic DNS services can do this, even if your IP changes from time to time. Just a few of the many available services:
  - http://dyndns.org
  - http://tzo.com
  - http://easydns.com
  - http://no−ip.com
  - http://www.microtech.co.gg/dns
- ADSL Deployment 'round the World Claims to have a complete list − looked accurate for my area − gives providers, prices, speeds, etc.
- comp.dcom.xdsl FAQ. Actively maintained, and a great technical reference for DSL technologies.
- comp.dcom.xdsl, DSL discussions, vents, and flames on Usenet. Good place to get technical questions answered that your ISP can't.

8.2. Glossary

A dictionary of some of the jargon used in this Document, and in the telco and DSL industries.

ADSL

Asymmetric Digital Subscriber Loop. "Asymmetric" in that the downstream potential is greater than the upstream. ADSL is capable of sharing on a single POTS wire pair. Maximum speed is 8 Mbps, though typically is limited by the provider to lesser speeds. The most popular DSL at this time.

ANT

ADSL Network Termination (a.k.a. the ADSL modem).

ARP

Address Resolution Protocol. Converts MAC addresses to IP addresses.

ASAM

Alcatel's terminology for a DSLAM.

ATM

Asynchronous Transfer Mode − provides high−speed packet switching from 155 Mbps to (currently) 2Gbps. Used to provide backbone switching for the Internet, and by many telcos since it can carry both voice and data. This is a common transport protocol for many telco DSL networks.

ATMF−25Mbps

ATM Forum Interface − 25Mbps speed, provided by a PCI NIC card. One of the interfaces used between the modem and PC.

brouter

A combination DSL modem that can be configured to act as either a bridge or a router.

CAP

Carrierless Amplitude Phase. A proprietary ADSL line encoding technique, that is (or was) in competition with "DMT". DMT has won the standards battle. CAP and DMT modems are not compatible with each other.

Central Office, or CO

Usually refers to one of two meanings: 1) The local Telco building that houses telephone equipment, and where local loops terminate. 2) The Telco voice switch that provides dial tone. Often referred to as just "CO". Typically, the CO houses one or more DSLAMs that make DSL possible. But, increasingly, DSLAMs are being deployed remotely.

CLEC

Competitive Local Exchange Carrier. "Competitors" to the ILECs. They do not own any lines, and must lease their lines from ILEC in order to provide any service.

CPE

Customer Premises Equipment − The Telco term for customer owned equipment (i.e. the stuff you are responsible for fixing). Examples are CSU/DSU, modems, fax machines, and your phone.

DHCP

Dynamic Host Configuration Protocol − A protocol used to distribute dynamically assigned IP addresses and other important networking parameters. The DHCP server "leases" an IP from its pool to clients on request. The lease is renewed at regular intervals. This is a common protocol on bridged DSL networks, and cable modem networks.

DMT

Discrete Multitone Technology. This is a line encoding common among ADSL deployments, and now is the standard. Sometimes referred to as "Alcatel compatible". Most telcos in the U.S. are now standardizing on DMT. The other, less common, ADSL encoding is "CAP". CAP and DMT modems are incompatible with each other.

DS0

The basic digital circuit for Telcos − offered at 56 Kbps or 64 Kbps. Can support one analog voice channel.

DSLAM

Digital Subscriber Loop Access Multiplexer − The Telco equipment installed at the CO that concentrates and multiplexes the DSL lines. One end of the copper loop connects to the DSLAM, the other to your modem. The DSLAM is essentially what makes DSL work. Increasingly, smaller devices that perform similar functions, are being deployed in remote locations in order to extend the reach of DSL.

DSL

Digital Subscriber Loop − A term describing a family of DSL services, including ADSL, SDSL, IDSL, RADSL, HDSL, VDSL, SHDSL, etc. that enable high speed Internet connections.

G.DMT

Synonymous with "full rate" ADSL. Used to distinguish between full rate ADSL, CAP based ADSL and G.Lite. See DSL Family for more.

G.Lite

A lesser version of ADSL that has lower maximum speeds, and requires no splitter or filters. Not DMT compatible. See DSL Family in this HOWTO for more.

HDSL

High bit rate DSL. See DSL Family in this HOWTO for more.

ILEC

Incumbent Local Exchange Carrier. The Regional phone company that physically owns the lines. Examples: Bell Atlantic and Pacific Bell. FCC regulations are forcing the ILECs to open up their networks to independent providers. This is allowing the independents like Covad and Rhythms to offer competitive services. This is a good thing for consumers IMHO.

ILEC

Incumbent Local Exchange Carrier. The Regional phone company that physically owns the lines. Examples: Bell Atlantic and Pacific Bell. FCC regulations are forcing the ILECs to open up their networks to independent providers. This is allowing the independents like Covad and Rhythms to offer competitive services. This is a good thing for consumers IMHO.

Interleaving

Interleaving is a tunable aspect of DMT/ADSL line encoding. It essential controls the 'interleaving' of bits in the transmission, and is used as a form of error correction. As interleaving increases, so does stability of marginal lines. It also increases latency.

IP

Internet Protocol. Often used to simply refer to an IP address.

ISP

Internet Service Provider. Even full−time connections require an ISP to provide basic Internet services and connectivity.

LAN

Local Area Network. A network of computers that are segregated from the WAN (Wide Area Network, i.e. the Internet). Often using private, non−routable IP addressing, e.g. 192.168.1.1 or 10.0.0.1.

Loop

The two wire twisted pair from the telco Central Office that terminates at a customer location. For DSL, a "clean" copper loop within the distance limitations is required.

MAC Address

Media Access Control Address. Sometimes also called "hardware" address, it is a unique identifier of network devices and is an important aspect of some network environments.

mini−RAM

Remote Access Multiplexer, a mini DSLAM. Typically with very few connections −− eight is common. Used for remote areas too far from a CO.

MTU

Maximum Transmission Unit, the largest packet size, measured in bytes, that a network can transmit. Any packets larger than the MTU are divided into smaller packets before being transmitted.

NAT

Network Address Translation is a means of allowing computers on a LAN to access the WAN while "masquerading" with the IP address of a host with a suitable address and configuration. With Linux this is called "ip−masquerading". Often used to share one public, routable IP address among hosts located on a LAN behind a masquerading proxy where the local addresses are private and non−routable.

NID

Network Interface Device − The telco housing on the side of your house. Typically where the telco's responsibility ends, and the owner's begins. Also, sometimes called the "SNI", "TNI" or "ONI" or other descriptive acronyms.

NIC

Network Interface Card − An internal PC card that supports the required network interface. Often an ethernet 10/100baseT or an ATMF−25Mbps card in this context.

NSP

Network Service Provider. An ISP's upstream provider or backbone provider.

OC−3

A fiber optic line capable of 155 Mbps.

POTS

Plain Old Telephone Service − The service that provides a single analog voice line (i.e. a traditional phone line).

PPPoA

PPPoATM, or Point−to−Point Protocol over ATM (RFC 2364). One of the PPP protocols being used by some DSL ISPs. This is really a device specific driver. A hardware device, i.e. a combination modem/router, is one alternative if this is the only option available to you.

PPPoE

Point−to−Point Protocol over Ethernet (RFC 2516). Another PPP protocol in use by providers. This one is more common, and there are several Linux clients available. See the Links section for more.

PPPoX

Used to refer to PPPoE and PPPoA collectively.

RADSL

Rate Adaptive DSL. See DSL Family in this HOWTO for more.

RBOC

Regional Bell Operating Company. The "Baby Bells". The U.S. phone companies that have had a state sponsored monopoly since the break up of AT&T.

RFI

Radio Frequency Interference. DSL is susceptible to RFI if in the right frequency range, and if close enough to the DSL signal. This can disrupt and consequently degrade the DSL signal. Unfortunately, DSL seems to operate in the frequency range of quite a few potential disrupting influences.

RWIN

Shorthand for 'Receive Window', aka the TCP Receive Window, a tunable aspect of TCP network stacks.

SDSL

Single Line DSL. Or, sometimes also "Symmetric DSL". See DSL Family for more.

SNI

Subscriber Network Interface − The Telco term for the phone wiring housing on the side of your house. It designates the point between the Telco side and the Inside Wire. This is also called the Demarcation Point. Sometimes called a "NID" also.

Splitter

The passive device (low−pass filter) at or near the NID that splits the DSL signal into separate voice and data channels. Filtering is required for most DSLs that share a POTS line.

Splitterless

A DSL installation that does not require a splitter. For higher speeds, a RJ11 filter (sometimes called microfilters) is placed on every extension phone jack where an analog phone or other non−DSL device is used, thus filtering the DSL signal at the jack, rather than at the NID. For lower speeds, no filter is necessary. Without a filter or splitter, the DSL signal tends to cause audible interference on voice phones. G.Lite needs no splitter, nor filter, but this is the exception to the rule.

SOHO

Small Office HOme

Sync Rate

The speed as negotiated by the DSL modem and the telco's DSLAM. This represents the theoretical maximum speed of the connection before any networking protocol overhead is taken into account. Real world throughput is always something less than the modem's sync rate.

T−DSL

German Telekom's ADSL implementation. See DSL Family for more.

T1

a.k.a DS1 − A digital dedicated line at 1.544 Mbps comprised of 24 channels, used for both voice (24 DS0s) and data.

T3

a.k.a DS3 − T1's big brother, a digital dedicated line at 44.736 Mbps, used for both voice (672 DS0s or 28 DS1s) and data.

VPI/VCI

VPI is "Virtual PATH Identifier" and is part of an ATM cell header. VCI is "Virtual Circuit Identifier", also part of an ATM cell header which contains circuit information. Technically speaking, these are really remote VPI and VCI (RVPI, RVCI). They are both important configuration aspects for modems and routers attached to ATM networks (the most common approach). They must match what the provider is using. Frequently used VPI/VCI pairs are 0/32, 0/35 and 8/35.

VDSL

Very high bit rate DSL. See DSL Family for more.

VoD

Video on Demand.

VoDSL

Voice over DSL.

WAN

Wide Area Network, a large publicly accessible network. For example, the Internet.

xDSL

Used to refer to the entire DSL family of related technologies: ADSL, SDSL, IDSL, etc.

8.3. Other Consumer Class High Speed Services

8.3.1. Cable Modem vs DSL

The Telcos see DSL as a competitor to the Cable Company's Cable Modem, and as such, are providing competitive pricing and configuration offerings. Although Cable Modems are advertised as having 10−30Mbps potential bandwidth, they use a shared transmission medium with many other users on the same line, and therefore performance varies, perhaps greatly, with the amount of traffic, time of day, and number of other users on the same node.

It is often heard that DSL has an advantage in that it is a private pipe to the Internet, with dedicated bandwidth. This is mostly a myth. You do have a private pipe to the DSLAM, but at that point, you enter the telco's ATM (or frame relay) network, and start sharing bandwidth. You are at the mercy of how well your DSL provider and ISP manage their networks. The consensus seems to be that DSL providers and ISPs are mostly doing a better job of managing bandwidth than the Cable companies. It is easier for them to add and adjust bandwidth as needed to meet demand. You are less likely to have speed fluctuations due to other users being on line at the same time. But, again, this gets down to how well the network and bandwidth are managed.

DSL probably has a small security advantage too. With most Cable modem networks, it is like being on a big LAN. You are sharing your connection (and bandwidth) right at the point of connection. But if you are not doing something to filter incoming connections already, you are asking for trouble either way.

There also seems to be a better chance of having ISP alternatives with DSL than Cable. At least, in the U.S. this is true. Choice is a good thing, and so is competition. It seems most Cable outfits give you just one choice for an ISP. If you don't like it, you are out of luck. The number of options with DSL probably varies greatly by geographic areas. Populous areas, like Northeast U.S., seem to have many options.

So which is better? The differences aren't as much with the technology, as they are with the implementations. If you look around, you can find plenty of horror stories on either. And plenty of happy customers too. The way to know what may be the best for you, is to do comparative shopping based on experiences of other users in your area. Don't base your choice on one person's opinion. This is statistically invalid. Likewise, don't base your choice on someone's opinion who has had a particular service for only a short time. Again, statistically not worth much. Get as many opinions from those that are using the *exact same services* that you are looking at.

8.3.2. Fiber in the Loop (IFITL or FTTC, and FTTH)

In some areas, newer neighborhoods are being built with fiber optic cable instead of the traditional telco copper lines. While the fiber is a definite problem for DSL services, it has it's own potential advantages. Existing fiber is potentially capable of 100 Mbps, and it looks like this could easily go up soon.

So while telco fiber customers are being shut out of the DSL market (since DSL is a copper only technology), they may have much to look forward to. Technologies are under development, and in some cases just now being deployed, to take advantage of fiber telco phone loops. Known as "FTTC" (Fiber To The Curb), or "IFITL" (Integrated Fiber In The Loop), this technology is another high speed service that telcos can offer. The speeds are sufficient for VoD (Video on Demand) and VoDSL (Voice over DSL), and other high bandwidth services. One nice advantage here is, that since there is no DSL signal on the wire, the only required CPE is a network card. In other words, no modem −− just connect a NIC to the wall jack and off you go! This will also allow the telco to provide other digital services such digital TV.

FTTC is Fiber To The Curb. The last leg into the house is still copper. FTTH (Fiber To The Home), on the other hand, is an all fiber loop with even higher potential.

8.3.3. Wireless

There is a lot of buzz about wireless technologies these days. Wireless would certainly seem to have a place in the broadband market, especially for areas that don't have ready access to cable or telco networks. There are still some inherent problems with the current state of this technology that may prevent it from becoming a major player in the near term however. Weather can still impact the wireless signal −− heavy cloud cover or rain for instance. Also, there is some pretty hefty latency if the uplink is via satellite. Surely these drawbacks will improve over time. But how soon?

8.4. Compatible Modems

This list is limited to those modems and delivery systems that are readily available, and should work with any current Linux distribution without having to go to extraordinary lengths. Alpha and Beta projects are not included.

Ethernet Interface

- *All* external, ethernet based modems, and modem combination devices, will work (provided they match the provider's DSL). The only requirement is a compatible ethernet network card. This is the preferred way to go.

PCI (Internal)

- Xpeed X200 IDSL http://www.xpeed.com/Products/x200/x200_c.html (as of kernel 2.2.18)
- Xpeed X300 SDSL http://www.xpeed.com/Products/x300/x300_c.html (as of kernel 2.2.18)

USB

- Alcatel SpeedTouch USB (ADSL): http://www.alcatel.com/consumer/dsl/supuser.htm#driver. The driver is kernel module and requires a 2.4 kernel.

8.5. Linux Friendly DSL ISPs

By "friendly" we mean ISPs that don't put up any unnecessary impediments just because you aren't running that other guy's OS. And yes, there is some of that going around. If your choices are limited, and you are forced to deal with one of these, then having a Windows box available temporarily is one work around. Another, may be to sweet talk the installer into letting you finish the installation (NIC, etc). Of course, self installation, if available, should be completely "Linux compatible".

So to make this list, the ISP/provider must make available some type of workable modem (ethernet interface at this point in time), nor should they penalize you, or make things difficult, just because you are running an alternate OS. Installing directly onto Linux should be an available option, and should not cause you any undue hardship. Technical support for Linux is a nice bonus, but not necessary to make the list. Please do not take these as recommendations, do your own homework. Also, this market is in a constant state of flux, so use this as a starting point only!

To add a name to this list, mail Linux_Friendly. Please included ISP's official name, URL (if not obvious), location and coverage area, modem type, server policy, and any other pertinent details.

National ISPs (U.S.):

- Speakeasy.net: Static IP and no PPPoX, servers explicitly allowed. Highly rated. National. Multiple IPs available.
- DirectTV DSL (formerly Telocity): Static IP, no PPPoX, liberal server policy. Reports of abysmal tech support. National. They have their own proprietary modem, but it is ethernet based.
- Penguinista DSL, DSL with a twist. Not just Linux friendly, but Linux lovers. Sponsored by the Benevolent Penguin Society. National. Static IP available. "Theoretical" timeouts and session limits though. Encapsulation protocol (PPP?) unknown. ???

Regional and Local ISPs (North America):

- qx.net, Lexington, Ky., and areas of Central and Eastern KY. Officially supports Linux. Static IP. Personal servers allowed. Tiered pricing plans. Highly rated.
- Commonwealth Technical Services, Richmond, Va. Officially, and happily support Linux. Static IP. Personal servers allowed. No bandwidth restrictions. This ISP runs on Linux!
- ExecDSL, Baltimore, MD, Washington, DC and surrounding areas. Static IP. Servers are OK. Various plans and DSL providers. Secondary MX and DNS available (nice touch!). (Apparently no official Linux support.)
- Netexpress.net, Moline, Ill. Tiered pricing. Static IP available. Apparently, no official support. Runs on Linux!
- iglou.com, Lexington, Ky., and soon in Louisville, Ky, Cincinnati, OH, and maybe Nashville, TN. Static IP available. Personal servers allowed. Tiered pricing plans with various options.
- Bluegrass.net, Lexington, Ky., and surrounding areas. Static IP. Personal servers allowed. Tiered pricing plans. Business class DSL only is available in Louisville, Ky.
- Netsync.net, Chautauqua County, NY (Fredonia, Jamestown, and surrounding areas). Static IP available, PPPoA, servers are OK. Linux is supported!
- Aracnet, greater Seattle, WA., and Portland and Salem, OR. areas. Static IP. Linux friendly! Tiered pricing. Shell access account is included (RH)!
- Drizzle.com, greater Seattle, WA area. Static IP, servers OK.
- Blarg! Online Services, Inc., greater Seattle, WA. area. Static or dynamic IP, PPPoA or Bridged connection. Personal servers allowed (no DNS or mail). Runs on Linux, and supports Linux!
- ReedMedia.net, Portland (Oregon) and surrounding areas; and surrounding areas of the following: Vancouver, Olympia, Tacoma, Seattle, Everett, Mt. Vernon, Bellingham (Washington). Various modem options, static IP available, personal servers are allowed.
- MM Internet, Southern California. Static IP, personal servers allowed, and secondary MX and DNS (nice!).
- Arrival.com, Central California. SDSL, servers allowed.
- DSLExtreme.com, greater Los Angeles area. Static IP, personal servers allowed.
- Bell Canada's Sympatico High Speed Edition. PPPoE.

European ISPs:

- Easynet Belgium. Linux is officially supported (Roaring Penguin). Dynamic IP.
- BaByXL Broadband DSL, the Netherlands. Linux is supported on Bridged/DHCP connections.

Other:

- iPrimus Pty Ltd, Sydney and Melbourne, Australia metro areas. Static IP, and multiple IPs available.

8.6. Setting up Linux as a Router

Depending on your local setup, you should consider some other issues. These include a firewall setup, and any associated configurations. For my setup, shown in Figure 5 below, I use an old i486 machine configured as a firewall/router between the DSL connection and the rest of my home network. I use private IP addresses on my private LAN subnet, and have configured my router to provide IP Masquerading and Firewalling between the LAN and WAN connections.

See the IP_Masquerade HOWTO , and Firewall_HOWTO for more information. For 2.4 kernels see the Linux 2.4 Advanced Routing HOWTO. My experience is that Linux is more flexible and provides superior routing/firewalling performance. It is much less expensive than a commercial router −− if you find an old 486 machine that you may be using as a doorstop somewhere.

Figure 5: A typical SOHO Network Setup

```
 <--Private Subnet/LAN-> Linux <-----ISP's Public Subnet----><--inet-->
      192.168.1.0


 X--+   --------
    |   |      |        --------      (eth0:0)---------
    +--=| Hub/ |       | Linux  |     +------=|  DSL   |=-DSL-> ISP's
 X-----=|Switch|=-----=| System |=----+       | Modem |       Gateway
    +--=|      |  eth1 |(Router)| eth0       ---------
    |   --------    |   --------    |
 X--+               |   IP_Masq     |
                    |  IP_Firewall  |
   |                |    Gateway    |
   |                |               |
   |                V               V
   V           192.168.1.1         Dynamic or
 192.168.1.x   LAN Gateway         Static IP
LAN Addresses  IP Address          from ISP pool
```

What I did is setup a Linux router (Redhat Linux 5.0 on a i486) with two ethernet interfaces. One interface routes to the ISP subnet/gateway (eth0 in above example), and the other interface (eth1 above) goes to a hub (or switch) and then connects the LAN with private network addresses (e.g. 192.168.1.x). Using the private network address behind your router/firewall allows some additional security because it is not directly addressable from outside. You have to explicitly masquerade your private addresses in order to connect to the Internet from the LAN. The LAN hosts will access the Internet via the second NIC (eth1) in the Linux router. Just set their gateway to the IP address of the second NIC, and assign them addresses on the same network.

Caution Make sure your kernel is complied with IP forwarding and the IP forwarding is turned on. You can check this with '**cat /proc/sys/net/ipv4/ip_forward**'. The value is "1" for on, and "0" for off. You can change this value by echoing the desired value into this file:

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

You will also need to set up "IP Masquerading" on the Linux router. Depending on your kernel version, this is done with **ipfwadm** (2.0), **ipchains** (2.2), or **iptables** (2.4). See the documentation for specifics on each. AND −− do not forget to have that firewall set up too!

There are also several projects that are devoted specifically to using Linux as a router, just for this type of situation. These are all−in−one solutions, that include security and various other features. Installation and configuration, is reportedly very easy. And these will run on very minimal hardware −− like a floppy drive only. The best known is http://www.linuxrouter.org. You might also want to look at http://www.freesco.org and http://www.coyotelinux.com. There is also http://www.clarkconnect.org/index.html, which is a similar concept but designed to be monitored and configured with a set of Windows based utilities.

9. Appendix: The Alcatel SpeedTouch USB ADSL Modem

Author: Chris Jones <chris@black−sun.co.uk>, copyright © 2001

with minor edits by Hal Burgiss <hal@foobox.net>.

9.1. Introduction

Installation, configuration and use of the Alcatel SpeedTouch USB ADSL modem with Linux.

French translation available
http://www.linuxdude.co.uk/docs/Alcatel−Speedtouch−USB−mini−HOWTO−FR.txt.

9.1.1. Preface

This is an adaptation of Chris Jones' Alcatel−Speedtouch−USB−mini−HOWTO. The current version of this document can be found at http://www.linuxdude.co.uk/docs/Alcatel−Speedtouch−USB−mini−HOWTO/.

As always with Linux, there is more than one way to skin a cat. This document takes one approach that is a working model. There are other ways, and there may be some contradictions to the `INSTALL` instructions as included by Alcatel. One notable example: The Alcatel driver supports both PPPoE and PPPoA. Alcatel suggests PPPoE as easier, but many are finding PPPoA more stable even if maybe a little more work to get going. This HOWTO only covers the PPPoA driver! The docs included by Alcatel have some information on PPPoE driver installation.

Also, this driver has just recently been released as of March 2001. It is still dependent on experimental code. Do not expect it to be as stable as ethernet modem solutions. Hopefully, this should change quickly as more users are able to provide feedback to the developers.

The driver also requires a 2.4.x kernel. If you are now running a 2.2 kernel, this will require a major upgrade −− of not just the kernel itself, but related packages such as modutils as well. You can find what all is required by reading `/usr/src/linux/Documentation/Changes` in the 2.4 kernel source tree. I would further suggest that you get a plain vanilla 2.4 system up and running stably before attempting the process below.

Warning
This process is not for the faint of heart. Novice users should strongly consider an ethernet modem, if this is a possibility.

9.1.2. Background

The Alcatel SpeedTouch USB ADSL modem has become a popular choice for providing low cost, widely available ADSL services, particularly in the UK, where BT Ignite (a division of British Telecom) install it for use over their ADSL network. Although BT plan to move to a new ADSL router soon, the SpeedTouch USB is still used by a large number of people, many of whom wish to use Linux, rather than Windows to access the Internet.

Initially the SpeedTouch USB was only useful to people running either Windows 2000 or Windows 98. Alcatel recently released a MacOS driver and have now released a Linux driver.

This document will show you, as simply as possible, how to configure a Linux machine to connect to the Internet using a SpeedTouch USB.

9.1.3. Assumptions

Several assumptions are made by this document:

1. You are using Red Hat 7.0 (although the instructions are very similar for other distributions). You should also have all the latest updates.
2. You are running the 2.4.2 kernel (this is not the appropriate place for kernel installation instructions; they are all over the web).
3. You are in the UK (this is the only ADSL network I have access to, so it's all I know. If you have any differences for your country, please email me and I'll amend the HOWTO).

The first assumption is merely because I use Red Hat 7.0 and so that is what I've had to use to get it working. I would like to cover other distributions, so if you spot any differences, please email me and I'll amend the HOWTO.

The second assumption is far more critical than the first − you MUST be running AT LEAST 2.4.1, but I have found 2.4.2 to be more stable and the SpeedTouch USB driver works fine with it.

9.1.4. Caveat

Unfortunately, much of the software required for this setup is beta quality and you may experience crashes, random oopses, etc. Hopefully work on the driver will bring its stability up to scratch and everyone can be happy.

9.2. Patching the kernel

For this section you will need the following items. (Please note that by the time you read this, the versions may have changed, so adjust accordingly.)

1. A working 2.4.2 kernel source tree (usually held in `/usr/src/linux/`). You should already be running the kernel − there are many guides on the web that will teach you how to install the 2.4 kernel.

2. The pppoatm kernel patch by Jens Axboe. This is available from:

   http://www.kernel.org/pub/linux/kernel/people/axboe/PPPoATM/2.4.1−pre7/pppoatm−1.gz

   To extract the patch from this, run **gzip −d pppoatm−1.gz**

3. Originally we used a combined kernel patch that covered the SpeedTouch driver and the SAR library it depends on. Unfortunately this patch has suffered some bit rot and isn't based on the current driver versions anymore so this next step is mildly more complex than it used to be.

   If your current kernel is using the combined patch (this will only be the case if you have followed this HOWTO before) and you want to upgrade to the latest versions of the SpeedTouch driver and SARlib you will need to 'revert' the combined patch (ie re−apply it to the kernel as you did originally, but with the '−R' option) and follow the SpeedTouch/SARlib instructions below.

   Firstly you need the latest SpeedTouch driver which is available from:

   http://sourceforge.net/

   You will also need the latest version of SARlib, available from:

   http://sourceforge.net/

   Extract both of the files you just downloaded with these commands:

   tar −xvzf sarlib−x.y.z.tgz
   tar −xvzf speedtouch−x.y.tar.gz

   Over−helpful browsers: It is possible that your web browser has already unzipped the above patches (i.e. Netscape automatically unzips .gz files). If this has happened, simply rename the patch files to remove the .gz suffix and continue with the following instructions.

Once you have all of these pre−requisites, we can start patching.

Patching Procedure

1. Change directory into the kernel source directory: **cd /usr/src/linux**
2. Test the `pppoatm−1` patch: **patch −p1 −s −E −−dry−run < /path/to/pppoatm−1**

   If this command produces any output then something is wrong with your 2.4.2 kernel source tree. Please fix it before continuing.

3. Apply the `pppoatm−1` patch: **patch −p1 −s −E < /path/to/pppoatm−1**
4. Now we need to compile the driver. This is very easy and can be done with the following commands:

   cd SpeedTouch (this directory is created by extracting the `speedtouch−x.y.tar.gz` archive)
   make
   make install (this needs to be done as root)

This will put the resulting kernel driver (`speedtch.o`) in `/lib/modules/2.4.x/kernel/drivers/usb/` (where 'x' is the kernel version you are running).

9.3. Configuring the kernel

For this section you will need:

- A working 2.4.x kernel patched according to the instructions in Patching the kernel

9.3.1. Configuration

To configure the kernel, load your favorite kernel configuration method: (e.g. **cd /usr/src/linux; make xconfig**)

You need to select the following kernel options:

- Code maturity levels−>Prompt for development and/or incomplete code/drivers
- Networking options−>Asynchronous Transfer Mode (ATM)
- Network device support−>PPP (point−to−point protocol) support
- Network device support−>PPP support for async serial ports
- Network device support−>PPP Deflate compression
- Network device support−>PPP BSD−Compress compression
- Network device support−>PPP over ATM
- USB support−>Support for USB
- USB support−>Preliminary USB device filesystem
- USB support−>UHCI (Intel, PIIX4, VIA, ...) support

  You may need to choose the OHCI driver here, depending on your USB controller)
- USB support−>Alcatel Speedtouch USB support

  This option is only available if you are using the older combined kernel patch for the kernel SpeedTouch driver. If you downloaded the separate SpeedTouch and SARlib tarballs then this option won't be shown and you don't need to worry about it.

  Selection: For simplicity, I recommend selecting **Y** (rather than **M**) for all of these, except the last two (UHCI/OHCI support and Alcatel Speedtouch USB support. Those are better left as **M**).

Now compile and install the kernel and modules as you would normally.

9.4. Installing the software

For this section you will need the items listed below. (Again, be aware of possible version updates.)

1. The results of Patching the kernel and Configuring the kernel.
2. Alcatel's binary management application. This is available from http://www.alcatel.com/consumer/dsl/dvrreg_lx.htm. You must download it from Alcatel, the license forbids us from distributing it.
3. A PPPoATM−aware PPP daemon. This is available from:
    - Red Hat 7 RPM (GLIBC2.2): http://sourceforge.net/project/showfiles.php?group_id=23818
    - Red Hat 6 RPM: NOT YET AVAILABLE
    - Debian .deb: http://sourceforge.net/project/showfiles.php?group_id=23818
    - Tarball: http://sourceforge.net/project/showfiles.php?group_id=23818
4. A working PPP setup (there are many guides on the Internet on how to configure PPP in 2.4 kernels)
5. The Linux HotPlug software from http://linux−hotplug.sourceforge.net. Install this as per its instructions (RPM installation is the simplest)

9.4.1. Installing pppd

Install whichever ppp package you need. You might have to create the file `/dev/ppp` yourself, which can be done with the following commands:

```
cd /dev
./MAKEDEV ppp
```

If your distribution does not include the **MAKEDEV** script, or it fails to create the correct device, you can use the following command:

```
mknod /dev/ppp c 180 0
```

9.4.2. Installing speedmgmt

Extract the Alcatel binary driver tarball (`speedmgmt.tar.gz`) and install it as per its included instructions.

9.5. Configuring the software

For this section you will need:

- The results of Patching the kernel, Configuring the kernel and Installing the software.

9.5.1. Configuring pppd

Edit the file `/etc/ppp/options` and replace its contents with the following:

```
lock
defaultroute
```

```
noipdefault
noauth
passive
asyncmap 0
name user@domain
user user@domain
plugin /usr/lib/pppd/plugins/pppoatm.so
0.38
```

Don't forget: You will need to replace the two "user@domain"s with your ADSL username.

Also, in the above example "0.38" is the VPI/VCI ATM pair for the author's provider. You will need to know what the correct values are for your provider, and substitute those. If these values are incorrect, you may sync, but will not be able to connect to your ISP's IP layer, and probably be frustrated. These values can be obtained from the Window's Alcatel client or your ISP. Other commonly used values: 0.32, 0.35, 8.32, 8.35.

Edit the file `/etc/ppp/chap−secrets` and replace its contents with the following:

```
# Secrets for authentication using CHAP
# client        server  secret                  IP addresses
user@domain     *       password
```

Now put the same contents in `/etc/ppp/pap−secrets`.

Don't forget: You will need to replace "user@domain" with your ADSL username and "password" with your ADSL password.

9.5.2. Final configuration

Note: At this point your ADSL modem should be *unplugged* from the computer.

1. Load the USB and Speedtouch kernel modules with these commands:
   a. **/sbin/modprobe usb−uhci**
   b. **/sbin/modprobe speedtch**
2. Mount the USB device filesystem with this command: **mount none /proc/bus/usb −tusbdevfs**
3. Start the Alcatel management application with this command: **/usr/sbin/mgmt**
4. Plug the ADSL modem into a USB port and wait a few seconds. It might be an idea to have a second terminal open watching `/var/log/messages` so you can see when the modem is initialized, which will look like this:

   Speedmgmt[2074]: Modem initialised at 576 kbit/s downstream and 288 kbit/s upstream

   Caveat: Some people have reported that plugging the modem in immediately before running the mgmt application works and can sometimes work better than plugging it in afterward.

   Once you have connected successfully, you might like to experiment with the sequence in which you perform the Final configuration. It is possible (and has been reported a number of times) to leave the modem plugged in continuously if you load the modules/apps in the correct order.

9.5.3. Connecting

If all of the software is installed and configured correctly, you should be able to initiate your ADSL connection with the command: **pppd**

Again, it would be useful to watch `/var/log/messages` so you can read pppd's output. Ideally you will connect, authenticate and be on the Internet. At this point you are on your own.

Assuming everything works you should see output from pppd similar to the following:

```
Mar 22 23:54:42 zanshin pppd[2076]: Plugin /usr/lib/pppd/plugins/pppoa
Mar 22 23:54:42 zanshin pppd[2076]: PPPoATM plugin_init
Mar 22 23:54:42 zanshin pppd[2076]: PPPoATM setdevname_pppoatm
Mar 22 23:54:42 zanshin pppd[2076]: PPPoATM setdevname_pppoatm − SUCCE
Mar 22 23:54:42 zanshin pppd[2077]: pppd 2.4.0b1 started by root, uid
Mar 22 23:54:42 zanshin pppd[2077]: Using interface ppp0
Mar 22 23:54:42 zanshin pppd[2077]: Connect: ppp0 <−−> 0.38
Mar 22 23:54:45 zanshin pppd[2077]: local  IP address xxx.xxx.xxx.xxx
Mar 22 23:54:45 zanshin pppd[2077]: remote IP address yyy.yyy.yyy.yyy
```

9.6. Known problems

The management application will most likely make your computer freeze when you try to reboot. There is a possible workaround for this − when your machine reboots, kill pppd, then mgmt.

As−is, a 2.4.2 or below kernel will crash if you try to use the Alcatel driver with an SMP build of the kernel. A patch is available that will fix this, from http://sourceforge.net/project/showfiles.php?group_id=23818

9.7. Revision History

- Version 1.1.3 − 16/06/01

```
Removed instructions for out−of−date combined kernel patch.
 (now replaced with instructions for using the official
  SpeedTouch and SARlib tarballs).
```

- Version 1.1.2 − 05/04/01

```
Added mention of SMP patch
```

- Version 1.1.0 − 25/03/01

  ```
  Converted to Docbook for better presentation
  ```
- Version 1.0.3 − 24/03/00

  ```
  Merged some changes from Giles Coochey
   regarding non−RedHat distributions
  Added link to French translation
   (courtesy of Vincent Besson)
  Some more minor tweaks
  ```
- Version 1.0.2 − 23/03/00

  ```
  Minor spelling tweaks from Bart King
  ```
- Version 1.0.1 − 23/03/00

  ```
  Some minor tweaks
  ```
- Version 1.0 − 23/03/00

  ```
  Initial release
  ```

9.8. Credits

Many thanks to all the people who send me ideas, corrections and additions for this HOWTO.

Special thanks must go to Telsa Gwynne for her outstanding help in the conversion of this document from a feeble HTML file to a glorious, spangly Docbook SGML file.