Ten years ago, most of us thought we would be able to live a full and happy life without worrying about whether we were getting maximum throughput across our networks, or whether the point-to-point latency on our machines would preclude us from popular gaming. But things have changed.

Televisions, games consoles and Linux machines all vie for IP addresses and bandwidth, usually on the same network, with poor wiring, poor layout and do-it-yourself support. Which is where we come in.

The Linux platform is the direct result of this network connectivity. It’s an operating system that was designed from the first line of the kernel to talk to another kernel, and as a result, it’s the perfect network troubleshooting platform. You won’t find settings hidden, parameters unprobed, or hardware unhinged under the surface, which is why Linux makes both a fantastic training ground for future system administrators, and a powerful ally in the hunt to track down intermittent problems and poor performance.

This is why we’ve pooled as many of the most common networking questions we could find. The answers should help you better understand how Linux handles the network, what kind of troubles are likely, how to get the best performance and how to create the most stable network in your neighbourhood.

“Linux makes a fantastic training ground for future system administrators.”
How can I change my hostname?

On the surface, this might sound like a rather technical question to start our networking article with. But your computer’s hostname is really just the name you give your machine when you install most distributions. Ubuntu will ask you for a descriptive name, for example, and this is displayed when you use a login screen or open a command prompt. But it’s also the name used for your computer on the local network, and this is why, historically, it’s called your system’s hostname – the machine that’s hosting your current session.

It’s also a perfect illustration of how difficult Linux can sometimes be to manage, because with most distributions, there’s no longer a GUI for changing something as simple as the hostname. The old Gnome network manager used to have a field you could change, but this has now gone, leaving users facing either the command line or a tool such as Ubuntu Tweak.

Fortunately, making the change on the command-line isn’t that difficult, but you do have a couple of choices. The hostname command should be up to the task, taking the new name as a single argument. But we found that this only seemed to make a temporary change on our system, which left us resorting to the old-school method: editing the /etc/hosts and the /etc/hostname files, and simply replacing the single occurrence in each of the old hostname with your new one.

Either way, after you’ve changed your hostname, you’ll need to logout and back in again to see the effect, and for seeing the changes across a network, it’s easier just to restart your machine.

How can I share my internet?

Now that most of us use a wireless router to access the internet, sharing an internet connection isn’t as much of an issue as it used to be. New machines can either connect wirelessly, or use a spare Ethernet port on the back of the router. But if you do find yourself wanting to share a connection from your machine, there are several different ways you can do so. If you have two Ethernet ports on your machine, one of which is connected to the internet, you can use the other port with a crossover cable to connect a spare machine. You then need to use your desktop’s network manager to enable sharing on the working connection.

Gnome’s default network manager includes this option, as does the version shipped with the latest versions of Ubuntu.

You also need to use the network manager if you’ve got two Ethernet cards and only want to enable one. If they’re both connected to valid DHCP servers (or are configured manually), and both use the same subdomain, they could conflict with one another. Fortunately, enabling and disabling connections through the network manager is as easy as either deleting it completely or selecting it from the drop down list of connections from the menu.

Enable network sharing in Ubuntu

1. Open connections
   Right-click on the networking applet, and then select Edit Connections from the pop-up menu.

2. Select new port
   Select the connection for the new machine, either in the Ethernet or wireless page, and click on Edit.

3. Share to others
   Switch to the IPv4 tab, and select Shared To Others followed by Apply. You may need to reboot to get sharing working.
How can I set my time using NTP?

Computers are really just big time pieces, yet despite their number crunching accuracy, they’re not the best devices for keeping time, especially when it comes to switching between daylight saving modes. And if you travel with a laptop, very few distributions detect a change in location and update the time accordingly.

Of course, if you’ve got the patience and a decent chronometer, you can easily update the time on your system yourself, using anything from the system BIOS to the little clock that sits in your task bar. But there’s a better way, and it’s a way that doesn’t require caffeine-enneged reflexes. You can use something called NTP: the Network Time Protocol. This will synchronise your clock with a couple of local servers, and in the process, turn your machine into a local atomic clock. Almost.

If you’re a GUI fiend, most system clock applications already include this feature. With the KDE desktop, for instance, you can right-click on the clock within the panel and select Adjust Date And Time. In the window that appears, you now just need to click on Set Date And Time Automatically. This will enable the Time Server field just below, and from there you should select an NTP server closest to your physical location.

If you prefer to use the terminal, the command you need to use is called ntpdate. It’s usually installed by default, but you’ll also need the internet addresses of two time servers. The easiest way to find them is to point a browser at www.pool.ntp.org and choose a couple of servers from the list on the right. Two are needed, because ntpdate triangulates the latency between you and the remote servers to generate a more accurate local reading. For this reason, it also helps if the two servers you choose are geographically close to your current location. It’s then just a matter of typing ntpdate -b 0.uk.pool.ntp.org 1.uk.pool.ntp.org, preceded by su or sudo if you don’t have the correct permissions to change the time. You’ll see the time update and the output from the command will tell you by how much the clock has needed to be adjusted, which can be a useful indicator of when you should next schedule a time update.

Why do websites sometimes fail to load?

If your network appears to be running properly, but typing a URL into a browser results in nothing but a timeout message, then there’s a good chance your problem is within a DNS server. It’s the job of the DNS to translate the text-based addresses we use for most servers and websites on the internet to the numeric IP address used by the hardware. If your DNS is working, for example, you can type ping linuxformat.com into a command line, and the first line of output should show something like the following:

```
PING linuxformat.com (80.244.178.150) 56(84) bytes of data.
```

If you’ve not used ping before, it one of the simplest network diagnostic commands you can run. It sends a ‘ping’ message to a remote server. If the message is received, the remote server sends a ‘ping’ message back. Believe it or not, the name comes from the sound a submarine’s sonar makes as it maps a surface.

One of the best things about changing the time on the command line is that you can see how far off your system clock is.
If it has, just replace the old address with the new one. If not, you could always replace the old DNS address with a public server, such as those run by OpenDNS and Google. OpenDNS addresses are 208.67.222.222 and 208.67.220.220. Google’s are easier to remember – 8.8.8.8 and 8.8.4.4 – but both promise to speed up the time it takes for your computer to get the IP address from a URL.

You can also change the DNS on a per-machine basis, and the graphical network configuration tool for your desktop should include the ability to modify the DNS server you’re using. If not, the file you need to edit is called /etc/resolv.conf. Newer distributions that rely on DHCP may not create this by default, so you’ll need to create the file yourself. It should just contain something similar to the following:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

As you should be able to see, we’ve used the IP addresses of Google’s DNS service, but you could easily replace these numbers with the ones for OpenDNS, or even your own router or gateway if that gives you the results you need. Either way, before you’ll feel the effect of any changes, you’ll need to either restart your machine, or the network (try service networking restart).

Why isn’t my USB modem working?

USB modems, of the old dial-up kind and the newer ADSL kind, are mostly dumb. All their functionality is loaded at boot time into the general-purpose processor running inside the box, and without a driver; these boxes can’t function. This is why so many USB modems don’t work with Linux, because it’s not a simple case of needing to reverse engineer the driver – it’s a case of developing all of the modem’s functionality from the ground up. The exceptions are when the firmware for the modem (that’s the code that runs on the processor) is made available. A Linux driver can then upload the code to the modem, which should then function perfectly. The only problem is that manufacturers are often reluctant to allow their firmware’s to be redistributed, and the binary redistribution of firmware irks many free software advocates.

If your modem isn’t working, this is likely to be the problem, and the best solution is something called NdisWrapper. This enables you to use Windows XP drivers for your device within your Linux environment. But before it will work, you have to first locate the .inf and .sys files hidden within the Windows driver package. The best way to tackle this formidable task is to head over to the NdisWrapper wiki at http://sourceforge.net/apps/mediawiki/ndiswrapper/index.php?mediawiki/index.php/List and search for your hardware. If your particular device is listed, you should find instructions on how to locate the files you require. It’s then just a case of using either the command line or the GTK GUI, launched by selecting Windows Wireless Drivers in Ubuntu’s Administration menu, can install new drivers with a couple of clicks.

Why is my network so slow?

You know the feeling. You’ve only got an hour to download the latest ISO of your favourite distribution and install it on your laptop before you go on holiday. But the internet isn’t your friend, and your high speed broadband connection feels more like a 56k modem from the dark days of 1999.

There are many potential bottlenecks in the network as data makes its way to your machine, and the first place to start is with the point where the internet joins your local network. For most of us, this is going to be a router either connected to your phone line using ADSL or a cable modem, and you can usually monitor the speed of the connection by connecting to the device through a web browser. The IP address you need to enter depends on your configuration, but typical values to try are 192.168.0.1, 192.168.1.1 and 192.168.1.254. You will probably also need to enter an IP address and a password. ADSL users should check the speed of their connection. This is usually expressed as an upload/download value in bytes, and it needs to be close to what you’re expecting. If not, then there’s likely to be a fault either in the quality of the service reaching your property, or within the property itself, such as a failed filter, interference or a broken phone line.

The best way to troubleshoot this connection is to plug your router into the master socket and check the speed again. Cable users can perform the same check with their modem, although problems are more likely to be the fault of the modem itself, rather than the cable coming into your property.

If connection speeds look good, the next place to look is your local network. One of your local machines may inadvertently be stealing the bandwidth, for instance, perhaps by running a BitTorrent client, or you may even be hosting a compromised Windows machine. The easiest place to check is on the router, but you’ll need an advanced model to view how much bandwidth is being used at any one time. The Fritz!Box we reviewed in LXF127 provides a graphical overview of bandwidth usage over time, which is useful if you want to check what’s happening overnight. But without a monitor on the router, a better option might be to remove all machines from the network and connect only your Linux box to see if the network is still slow.

You can change your DNS configuration on a per-connection basis, which is useful if you switch between Ethernet and Wi-Fi.

www.tuxradar.com

October 2010 LXF136 | 43
Networking made easy

» Is my ISP blocking or shaping my data?

A

Despite many ISPs advertising 'unlimited' broadband packages, very few offer what we’d call a genuinely unlimited service. The two main targets for compromise are port blocking and traffic shaping. The former may stop you from running your own mail or web server, for example, or from using VoIP for internet phone calls. The latter will selectively reduce the amount of bandwidth for specific services, such as peer-to-peer networks, video streaming sites and FTP.

The best way of checking to see if your ISP is blocking access to ports on your network is to perform a scan on your network from somewhere else. From a remote Linux machine, for example, you could type `nmap` followed by the IP address of your network. GUI users should take a look at `Zenmap`, and if you can’t get access to a machine outside of your domain, try one of the many websites that will scan an IP address for you, reporting on any open ports it finds. If you find that common ports are blocked (such as 80 for HTTP or 25 for sending mail using SMTP), but are open when you run the same scan from within the network, then your ISP is blocking access. You could try changing the port for those services: SSH and SFTP can easily be shifted using their configuration files. HTTP/Apache can do the same trick, but your users would then need to access your website by specifying the port manually rather than just typing the web address. If all else fails, you’ll need to ask your ISP to remove the restrictions, or move to another ISP.

Detecting whether your ISP is shaping traffic is harder than finding out whether ports are being blocked. This is because your internet speed will normally fluctuate at certain times of the day, or during periods of high demand, regardless of whether your ISP is throttling your network or not. The best indication is usually a bump in speed at the same time every night, which you can watch for if you’re sharing your favourite distribution through Vuze, for example. But this change may not always be obvious or at the same time or day, or it may affect a service, such as video streaming, where a visible difference isn’t so easy to detect. In these cases it’s best to check with your ISP’s quality of service statements, or ask them directly.

Side-stepping traffic shaping is like avoiding blocked ports. You need to make the problematic service look like something else.

Advanced network troubleshooting

If you’ve traced network problems back to your Linux machine, and you’ve ruled out a failing interface or broken cable, there are several more things to try. The first step should be to make sure your network really is slow. We’d recommend using a command like `wget` to grab an ISO of your favourite distribution, and see what speed you get. Canonical’s list of Ubuntu mirrors is always a good target.

The next step should then be to run the perennial `ifconfig` command from the terminal. At first glance its output looks a little intimidating, but it’s quite easy to decode when you know what you’re looking for. It’s essentially a list of every network device on your system, along with an overview of their current status. This status includes their IP addresses, if they have them, and the number of errors encountered by the protocol as it tried to make sense of the data it’s receiving and sending.

With the simplest setup, you’ll see only two devices, typically `eth0` and `lo`. The first is the default Ethernet port on your system, while the second is called the ‘loopback’ device. This is only for testing and convenience – it’s the virtual device that forwards requests for access to IP address 127.0.0.1 and `http://localhost` to servers running on your own local machine. You can see this by the IP address assignment if you run virtualisation software, such as VMware or VirtualBox; you’ll also see several other virtual devices, and these are used to create the network bridges between your real connection and those used by the virtual machine. If you use a wireless connection, this will be listed as `wifi0` or possibly `ath0`.

The important bit to check in the output from `ifconfig` is the errors, drops and overruns part. The more errors you see, the more likely the performance of your network is going to be affected. Errors with wireless networks will dramatically effect the transmission rate, for example, but wireless errors are inevitable as the signal passes from the router to your machine. Errors with `eth0` on the other hand, shouldn’t appear. If they do, you get can more in-depth information on network errors by installing the `ethtool` utility and executing `ethtool -s eth0` on the command line with administrator privileges.

A small collision rate is nothing to be worried about. If you’re getting lots, this is a symptom of too much traffic on your network. Late collisions (tx_late_collision and rx_late_collision) could be a sign that your network cables are too long. If you’re getting CRC (tx_crc_errors and rx_crc_errors) errors too, this might be an indication of physical problems with your Ethernet cables, with one of your ports or the card itself. Another common problem is detected by excessive frame align errors (rx_frame_align_error), and these are often caused by electrical interference somewhere over your data path.

You can also use `ethtool` to verify the speed at which your network card is running, which might also be a factor in network problems. Typing `ethtool eth0` for example, will produce output listing the capabilities of your hardware, as well as its configuration. You should see the speed listed at at least 100Mb/s; `Duplex: Full` and `Link Detected: yes`. If not, try switching your network card for an alternative device.
else, which invariably means changing the range of ports being used by the server. ISPs will also look at the kind of traffic being transferred, rather than just the ports, and you can scupper this tactic by encrypting the data between your network and the remote site. But this is still an avoidance tactic, and you’re still better off finding a better ISP.

Q: Why can’t I share my files?
A: If you’ve got more than one computer running on the same network, sooner or later you’re going to want to get a file from one to the other. Unfortunately, as you might have already noticed, this seemingly simple request can quickly become a nightmare. But it is possible to set up file sharing without having to resort to sending it to yourself as an email attachment. The key to this is something called Samba. This is the protocol used to negotiate the file moving from one machine to another, as well as convenient tasks like the remote browsing of files and folders, and even sharing a remote printer.

Samba can be installed and configured independently, but for ease of use and flexibility, we’d recommend trying to install it through your desktop environment first. And if you don’t have a preference, we’ve had better luck with Gnome than KDE, which seems to have a few problems enabling file sharing. With Ubuntu’s Gnome, for instance, you just need to right-click on the folder and select Sharing Options from the menu that appears. If these options aren’t there, install the nautilus-share package.

You then need to enable Share This Folder, after which the system will install a few packages to enable the service then ask to restart the session. When you’re able to edit the folder options again, decide whether to enable others to view the folder and click on Create Share. You should now see two small arrow emblems above the folder icon, denoting that file sharing is enabled. Remote users should now see your folder, and be able to access it, when the browse the local network.

KDE users will need to make sure they’ve got both the kdenetwork-filesharing and main Samba package, neither of which are installed by Kubuntu. You then need to right-click on any folder you want to share, such as Public in your Home folder, and click on the Properties option. From the window that appears, click on the Share tab, followed by the Configure File Sharing button. After entering your administrator password, you should see the File Sharing control module window. Now click on the Allowed Users button and select Allow All Users To Share Folders from the window that appears, and click on Apply. You should now be able to set per-folder sharing for the folder back in the Dolphin main window.

Q: Why can’t I access remote files?
A: Setting Samba so that other people can see your files is one thing, accessing other people’s files through Samba is quite another. Fortunately, it’s a much easier process, and doesn’t normally require you to install any further packages. From Gnome, for example, just select Network from the Places menu. In the file manager window that appears, you should see an icon for every Samba-running machine on your local network, and clicking on any of these will let you see specific files and folders that are being shared.

If the machine you want doesn’t appear, and you know its IP address, you can select Connect To Server from the Places menu and then select Windows Share as the service type. Just enter the IP address into the server field and click on Connect. A file manager will then appear, hopefully showing the files and folder you’re after.

KDE users need to open the Dolphin file manager, and can then browser the local network by clicking on the Network icon in the left-hand panel of the display. But you can also connect to a server directly by using a Samba URL in the form smb://username@server. If you omit the username, Dolphin will ask for both this and the password before opening the location within the current file browsing session. You can also use the same SMB URL within Gnome, if you enter the location manually, and within other applications such as Firefox, which will enable you to browse a Samba share and download files through the same browser window.

Firefox will let you browse a Samba share and download files through the same window.

Networking made easy

Samba shares can be accessed from Linux, Windows and OS X machines without any further configuration.
Most of us use SSH to connect to remote servers, using anything from a workstation to a games console. You may think that a tool whose acronym means ‘secure shell’, is already secure, and it is – for the most part. But there are some well-known vulnerabilities in SSH, which means that there’s a lot you can do to make it safer.

The easiest, and probably the most effective of these fixes, is to make sure only certain users can log in through SSH and that their passwords are secure. This is the weakest link in the chain, and if you have an account called ‘test’ with a password of ‘test’, it’s highly probable that before long someone will discover the weakness and gain access to your machine.

Another crucial step is to disable support for the older version, version 1, which has been successfully hacked in the past. This, and all the other configuration options for SSH, are hidden within its configuration file, which you can usually find within the `/etc/ssh` directory. Newer versions are called `ssh_config`, and you’ll need to open this with your favourite text editor with root privileges (we recommend Nano).

To disable support for the older version, look for the line ‘Protocol 2,1’. This is telling the server to support both versions 2 and 1 of the SSH protocol, and we need to remove the ‘1’ to disable support for the older. If this line starts with a `#` symbol, meaning it’s commented out, remove this too. While you’re playing with the configuration file, there’s lots more you can do to make your connections more secure. We’d recommend adding a line reading ‘PermitRootLogin no’, for instance, as this will disable anyone connecting directly to your system’s administrator account. If you need root access, you can always `su` to switch from an ordinary user, or execute commands with `sudo`.

Another security tip for people connecting to your SSH server is to add ‘MaxAuthTries 3’ to the configuration file. This will only let people attempt three connections to your server before being kicked. But perhaps the best change you can make if you’re worried about unauthorised access is to move the SSH server from the default port of 22 to something else. This is because port 22 is very well known, and there are thousands of scripts running across the internet looking for SSH on port 22. Simply switching this to something else immediately solves the problem, and removes your server from the probing eyes of an unscrupulous script. To change ports, look for ‘Port 22’ in the configuration file, and change this to something else. Alternatively, if you’re running a firewall on a router before your Linux machine, just get some random remote port to point at port 22 on your SSH server.

After you make any changes to the SSH configuration, you need to restart the server. Debian/Ubuntu users can do this by typing `service ssh restart` from the command line, or by simply restarting your system.

SSH is one of Linux’s best features, and it means you can access your machine from almost anywhere, from anything.
results of a domain name search that might include contact details if you fancy dropping the offender’s ISP an email.

**Q: Why isn’t Wi-Fi working?**

A: It wasn’t so long ago that Wi-Fi was a configuration nightmare on Linux. Even if you had working drivers for your Wi-Fi device, it was still an effort to get it installed, configured and running. But things have changed so much over the last couple of years that most Wi-Fi devices will configure themselves automatically and run without problems. You should be able to see the wireless signal strength indicator in your task bar, and switch between networks, enter passwords and stop the service with a couple of mouse clicks.

But there can still be problems. If your card isn’t recognised in the first instance, the tool you need is `iwconfig`. This is the wireless equivalent to `ifconfig`, performing the same diagnostic and configuration functions on wireless, rather than wired, networks. And the best place to start is by typing `iwconfig` on the command line. If your wireless device has been detected by the system, and the driver correctly installed, you should see a wireless device listed in the output. If not, you need to start looking at your system logs.

If you’re using a USB device, try plugging it in and typing `dmesg`. You should see your system trying to detect the hardware used by the device, and hopefully, loading the driver. If not, you will need to check for driver support for your specific hardware on your chosen distribution. But the hardware used by your device isn’t always obvious. It’s rarely the specification printed on the packaging or the device itself, for example. You need to know the exact chipset used by the hardware, and the best way to do this is by either typing `lspci` or `lsusb` on the command line when the device is connected, although you may also see the detected hardware listed in the output from `dmesg`. The output from our USB Wi-Fi receiver included the following:

```
0cf3:9170 Atheros Communications, Inc. AR9170 802.11n
```

A quick Google for AR9170 revealed that the driver for this device was part of the 2.6.31 kernel, and as a result, would require a distro using at least that version. As soon as we switched to a distribution using that kernel (Ubuntu 10.04), the device worked without any further configuration.

Another solution is to block these overzealous attempts to access your machine. You could do this manually, adding the offending IP addresses to the `/etc/hosts.deny` config file. But there’s an easier way, and this uses a tool called DenyHosts, which you’ll probably need to install from your system’s package manager. This is a great utility that tracks changes to your log files, automatically detects spurious connections, and blocks the offender’s IP address from further attempts if they pass various thresholds. After it’s installed, these thresholds can be modified by opening `/etc/denyhosts.conf` in your favourite text editor. We’d recommend reducing the `DENY_THRESHOLD_INVALID` value from 5 to 2. This is the number of failed attempts allowed when the remote server tries a username that doesn’t exist on your system. We’d reduce `DENY_THRESHOLD_VALID`, the parameters for attempts that do correspond to a valid username, to 3, which should give legitimate users a chance to access your system if they suffer brief password amnesia.

After editing the file, the daemon can be started by typing `service denyhosts start` as system administrator. The only danger now is that you lock yourself out of your own system. You’ll then either have to get physical access to your machine, or wait for the period specified by the `AGE_RESET_VALID` parameter, which defaults to five days. If you want to see how well DenyHosts is doing, check `/var/log/denyhosts`. Each address that has been blocked will be listed, along with the

---

**Networking made easy**

---

**KDE wireless connections**

1. **Open Networking**
   
   Click on the small plug icon in the applet toolbar, and a menu appears. Enable wireless, then click on Create Network connection.

2. **Choose a network**
   
   A window appears listing all the available wireless networks detected by your device. Select your network and click on Connect.

3. **Enter password**
   
   If your network is password protected, which all but open access points should be, enter the password and click on OK to connect.

---

Q&A: Open Networking

A: Why isn’t Wi-Fi working?

Why isn’t Wi-Fi working? It wasn’t so long ago that Wi-Fi was a configuration nightmare on Linux. Even if you had working drivers for your Wi-Fi device, it was still an effort to get it installed, configured and running. But things have changed so much over the last couple of years that most Wi-Fi devices will configure themselves automatically and run without problems. You should be able to see the wireless signal strength indicator in your task bar, and switch between networks, enter passwords and stop the service with a couple of mouse clicks.

But there can still be problems. If your card isn’t recognised in the first instance, the tool you need is `iwconfig`. This is the wireless equivalent to `ifconfig`, performing the same diagnostic and configuration functions on wireless, rather than wired, networks. And the best place to start is by typing `iwconfig` on the command line. If your wireless device has been detected by the system, and the driver correctly installed, you should see a wireless device listed in the output. If not, you need to start looking at your system logs.

If you’re using a USB device, try plugging it in and typing `dmesg`. You should see your system trying to detect the hardware used by the device, and hopefully, loading the driver. If not, you will need to check for driver support for your specific hardware on your chosen distribution. But the hardware used by your device isn’t always obvious. It’s rarely the specification printed on the packaging or the device itself, for example. You need to know the exact chipset used by the hardware, and the best way to do this is by either typing `lspci` or `lsusb` on the command line when the device is connected, although you may also see the detected hardware listed in the output from `dmesg`. The output from our USB Wi-Fi receiver included the following:

```
0cf3:9170 Atheros Communications, Inc. AR9170 802.11n
```

A quick Google for AR9170 revealed that the driver for this device was part of the 2.6.31 kernel, and as a result, would require a distro using at least that version. As soon as we switched to a distribution using that kernel (Ubuntu 10.04), the device worked without any further configuration.