ITS 332 Networking Lab

Networking Tools

Dr Steven Gordon

Revision 927

6 November 2009

1 Overview

This lab will introduce you to important software tools for managing computer networks. It will also give you an opportunity to become familiar with the ICT Networking Laboratory room, e.g. the computers, operating systems and network equipment. The software tools you learn in this lab will be used in the remaining labs in the course.

2 Background

In ITS 332 you will perform a variety tasks in configuring and managing computer networks, as well as developing and using computer network applications. There are various tools available on most computers that can be used to support these tasks. The tasks include:

- Viewing and changing the configuration of your computer's network interface, such as addresses and other protocol parameters.
- Testing your computer's network connectivity, such as ability to communicate with other computers and statistics of the communication.
- View and analyse traffic sent/received by your computer, as well as other computers on a network.

3 Operating Systems and Tool Interfaces

The tools that can be used to manage the network vary on different operating systems. For example, Microsoft Windows has different programs than Unix variants such as Ubuntu and Apple MAC OS. (And indeed, the programs may be different between versions: Windows XP may be different from Windows Vista, and Ubuntu Linux different from RedHat Linux). Combined with this, many operating systems will have two different interfaces to the same tool: a graphical user interface (GUI) and a command line (text) interface.

Although the programs may be different (including interface and options), the majority of them provide similar level of functionality. Therefore once you learn the functionality using one tool, it will not be too hard for you to perform the same functionality in another operating system.

For our lab classes, we will use Ubuntu Linux, for the reasons outlined in the Introduction to Ubuntu Linux. We will show examples and expect you to use the command line interface on most occasions. This is because once you know the command line interface, it is very easy to perform the same operations in the GUI (however, vice versa is not true: if you learn the GUI, it may be hard to understand the options of the command line interface). Also note that some network equipment is managed by a command line interface: e.g. you may log on to a router or switch and set the configuration via the command line interface only.

4 Viewing Network Interface Information

Your computer connects to the LAN via one of its Network Interface Cards (NIC) (see the overview of lab Facilities for details). In the Networking Lab, each computer has two Fast Ethernet NICs, and by default one of the NICs is connected to a Fast Ethernet switch (in the switching cabinet in the corner of the room). Almost all operating systems allow the user to view information about the current NIC connection, including:

- MAC (or hardware) address
- IP address and subnet mask
- Addresses of other important nodes (servers) on the network
- Traffic sent/received by the NIC

Operating systems often allow administrator users to modify some of the above information as well.

The main command to view and edit the network interface information is `ifconfig`.

Note 1. *(Microsoft Windows) The equivalent command to* ***`ifconfig`*** *on Microsoft Windows XP is* ***`ipconfig`****.*

To view the information for all interfaces:

```
ifconfig
```

Task 1. *How many interfaces on your computer? What device or software is each interface associated with?*

Interfaces are given names such as *eth0* and *eth1* for Ethernet network interface cards (remember, the PCs have two Ethernet cards), and *lo* for loopback address (a special interface to communicate with your own computer). To view the details of a specific interface, such as *eth0*:

```
ifconfig eth0
```

Task 2. *What are the following addresses for each of your Ethernet cards: hardware, IP, network, broadcast, subnet mask?*

5 Testing Network Connectivity

A basic task for diagnosing the connectivity of a network is to test whether one computer can communicate with another. This is normally performed using the Internet Control Message Protocol (ICMP). A user application that implements ICMP for testing connectivity is `ping`.

`ping` sends a message from your computer to some destination computer, which then immediately responds. `ping` measures the time it takes from sending the message, to when the response is received. That is, the delay to the destination and back, i.e. the *round trip time* (RTT).

The simplest way to use `ping` is to specify the destination as the first parameter:

```
ping DESTINATION
```

where `DESTINATION` is the IP address or domain name of the computer you want to test connectivity with.

Task 3. *What does the output of* `ping` *tell you if the destination is the computer next to you?*

You can stop the ping by pressing *Ctrl-C*, or you can limit the number of messages sent by `ping` to `COUNT` messages using the `-c` parameter:

```
ping -c COUNT DESTINATION
```

Task 4. *Are the results from the Task 3 any different from pinging the SIIT web server?*

5.1 ping at SIIT

`ping` is a very simple, but useful tool to diagnosing network problems. However, `ping` (and more generally, ICMP messages) can be used to cause problems in a network. For example, a malicious user may perform a security attack on a network by sending many ICMP messages to a router (making the router too busy to handle normal traffic, thereby restricting use of the network). Therefore, some organisations decide to not allow ICMP messages into and/or out of a network. SIIT does this: from inside the lab you cannot `ping` a computer outside on the Internet (e.g. try to `ping` `http://www.google.com/`). This is done for good reasons by the SIIT Network Administrators, however makes it difficult to demonstrate `ping` and other ICMP-based tools in this lab!

Task 5. *Try for yourself: what happens when you ping www.google.com? What about other web servers on the Internet?*

In addition to a network administrator blocking ICMP from leaving the network, some organisations may block ICMP from entering a network, and more specifically, block a particular computer from responding to ICMP messages. For example, the web server `www.fakewebserver.com` may be configured to not respond to ICMP messages, therefore your `ping` to such a domain would get no response.

Luckily for us, there are free web sites that allow us to use `ping` from the website to any computer that responds to ICMP messages. Note that when using these websites the source of the ICMP message is not your computer, but is the web server of the site or a router/server selected from the site.

There is an excellent list of free web-based `ping` (and other) tools at: `http://www.bgp4.net/wiki/doku.php?id=tools:ipv4_ping`. Several you should try include:

- Qwest Looking Glass Asia (`http://stat.qwest.net/looking_glass_asia.html`) - source is from Hong Kong, Singapore, Sydney or Tokyo
- AIT Network Test Tools (`http://www.cs.ait.ac.th/cgi-bin/Net/net-test`) - source is Thailand
- Cogent Looking Glass (`http://cogentco.com/htdocs/glass.php`) - source from cities in North America and Europe
- Telia Looking Glass (`http://lg.telia.net/`) - sources from cities in Europe
- Carnegie Mellon University Network Operations (`http://www.net.cmu.edu/cgi-bin/netops.cgi`) - source from US

Task 6. *Using any of the above websites, ping the SIIT web server from two different geographical locations. How does it compare with the results from Task 4?*

6 Testing a Route

Another useful network connectivity test is to determine the path (or route) that a message takes. That is, what routers does the message pass via on the way to the destination. As with `ping`, ICMP messages are sent to determine this. An application that implements this in Ubuntu is `tracepath`. Like `ping`, an ICMP message is sent to the destination and returned, but with `tracepath` the set of routers along the way also send a response to the source.

Note 2. *(Other Unix Applications) Some Unix distributions use the application* `traceroute` *to perform the same functionality as* `tracepath`*. In fact, you will see many web sites refer to* traceroute *instead of* tracepath*.*

Note 3. *(Microsoft Windows) The equivalent command to* `tracepath` *on Microsoft Windows XP is* `tracert`*.*

The application can be used by giving a destination IP address or domain name as a parameter:

```
tracepath DESTINATION
```

Task 7. *Examine the output from using* `tracepath` *to the SIIT web server. What useful information does it tell you?*

As `tracepath` uses ICMP, it suffers the same drawbacks on SIIT's network as `ping`. In some cases, you may get a *no reply* message from a router. But again, you can use the free web-based applications in Section 5.1 to demonstrate `tracepath` (often referred to as *traceroute*).

Task 8. *Use two different geographical locations from the free websites, find the path to the SIIT web server. How does it compare with the results from Task 7?*

7 Converting between Domain Names and IP Addresses

We know that the Domain Name Service (DNS) is used for mapping domain names (user-friendly addresses) into IP addresses (computer-readable addreses). It is also possible to do the opposite, often referred to as *reverse DNS*: map IP addresses to the corresponding domain name.

There are several tools for using DNS (or reverse DNS) in Ubuntu, all using slightly different approaches, and producing different output. The tools are: `nslookup`, `host` and `dig`. The basic use of the tools work in the same way: give a domain name as a parameter, and the corresponding IP address will be returned; or give an IP address as a parameter, and the corresponding domain name will be returned.

```
host DOMAIN         # returns IP address
nslookup DOMAIN     # returns IP address
dig DOMAIN          # returns IP address
host IPADDRESS      # returns domain name
nslookup IPADDRESS  # returns domain name
dig IPADDRESS       # returns domain name
```

For now, lets use `nslookup` since it is the most common of the three (also available on MS Windows).

Task 9. *Use `nslookup` to find the IP address for `www.siit.tu.ac.th` and `www.google.com`. Explain the output that is produced.*

Task 10. *Using the IP addresses found from Task 9, perform an reverse DNS lookup using `nslookup` to find the domain names. Are the results what you expected?*

By default, each tool will try to first use your local DNS server to retrieve the information. If you want to retrieve the information from a specific DNS server (such as `ns.siit.tu.ac.th` or `ns1.sprintlink.net`) then you need to give an additional option:

```
nslookup DOMAIN DNSSERVER
```

Task 11. *Perform Task 9, but this time using the DNS server* `ns1.sprintlink.net`. *How do the results compare with Task 9?*

Note 4. *(Hosts and DNS) Note that Linux typically uses (at least) two naming services: the common Internet naming service DNS, as well as a simple file that lists a set of names and corresponding addresses. This is called the* hosts *file. See Section 11 for further information.*

8 Viewing the Routing Table

IP uses routing tables to determine where to send datagrams. This applies to end hosts (like PCs), as well as routers, however a routing table on a host is typically quite simple, since all packets are often sent to a local (default) router.

You can view your routing table using the `route` command:

```
route -n
```

The `-n` option means the output will contain the numerical IP addresses (rather than the default domain names).

Task 12. *What do the first three columns in the output of* `route` *mean?*

Task 13. *Look at the output of* `route`*. Try to understand what each entry (row) may be used for. Write a simple explanation of each entry.*

By default, `route` shows the main routing table. However, the operating system also maintains a cache of routing entries, which are based on where previous packets have been sent. When IP has a packet to send, it first checks the routing cache for an entry, and then (if no entry exists in the cache) uses the main routing table. You can view the routing cache using the `-C` option:

```
route -n -C
```

The routing cache shows the Gateway used for particular Source and Destination pairs.

Task 14. *Look at your routing cache. Try to understand what the first 2 or 3 entries (rows) have been used for. Write a simple explanation.*

In a later lab we will use `route` to modify the routing tables (like adding a new route).

9 Converting IP Addresses to Hardware Addresses

Remember that IP addresses are logical addresses. For a computer to send data to another computer on the same LAN/WAN they must use hardware (or MAC) addresses. For example, if computer A wants to send an IP datagram to computer B (on the same network as A) with IP address 192.168.1.3, then computer A must know the hardware address of computer B. Hence, the Address Resolution Protocol (ARP) is used to find the corresponding hardware addresses for a given IP address.

Although we don't yet cover in detail how ARP works, we can view the information ARP has in your computer using the application `arp`. Running `arp` will return a table (called the *ARP table* or *ARP cache*) of IP addresses and corresponding hardware addresses that your computer currently knows about:

```
arp -n
```

Task 15. *List the IP address/hardware address mappings that your computer is aware of. What other information does* `arp` *tell you?*

ARP automatically updates the table with new entries for you. However, you can also use `arp` to delete entries from your ARP table and manually add new entries.

10 Network Statistics

A tool that allows you to view many different network statistics is `netstat`. For example, you can view interface statistics (similar to `ifconfig`), routing table statistics (same as `route`), connection statistics and TCP/IP packet statistics. Lets look at how to view the last two.

First, you can view the active TCP connections:

```
netstat -n -t
```

Task 16. *Visit your favourite web site. View the active TCP connections using* `netstat` *and explain what they show. That is, what does the Local Address and Foreign Address mean?*

You can also view summary TCP/IP statistics:

```
netstat -s
```

Task 17. *View and browse through the summary statistics using* `netstat`*.*

11 Viewing More Network Information: Useful Files

Some additional networking information about your computer can be found in various files on your computer. An important directory that contains a lot of configuration details for your operating system is the `/etc` directory. Some useful files include:

/etc/hosts Set a list of local domain names and corresponding IP addresses. Used in addition to DNS. Normally this would be used to give a name to your computer, as well as other computers on your network.

/etc/resolv.conf Indicates the local DNS server for this computer.

/etc/network/interfaces Stores information about your computers' network interfaces.

/etc/services List of port numbers and corresponding servers

Task 18. *From the above set of files, what is your: DNS server(s)? host name?*

12 Automatic IP Address Configuration

12.1 How Does It Work?

When an operating system is installed on a computer and the computer first setup (by, for example, the network administrator), the IP address and other relevant network information (such as DNS servers, subnet mask) can be manually entered. In Ubuntu, commands like `ifconfig` can be used to do this.

But with manual configuration, if any network information changes, the network administrator must then go to each computer to make the changes. With the SIIT Bangkadi network of 300 or more computers, the task of manually configuring each computer if, for example, the DNS server IP address changes, would be enormous!

Therefore, in practice there are ways to automatically configure a computers network information. The most used method is called *Dynamic Host Configuration Protocol* or DHCP. The basic process using DHCP is as follows:

1. One computer on the network is configured as a *DHCP Server*. This contains information about the possible IP addresses that can be allocated to other computers, and the DNS servers to be used. Usually, the DHCP Server is a router on the network.

2. All the hosts in the network are configured to use a *DHCP Client*. When the computers are first setup by the network administrator, no information about IP address, DNS server is given.

3. When a host boots, the DHCP Client broadcasts a request for an IP address. In other words the host sends a message to everyone else on the network saying: "I need an IP address (and other information)".

4. The DHCP Server is the only computer that responds: the DHCP Server selects an IP address for the host and sends it, including the network DNS server, subnet mask etc. to the host.

5. The DHCP Client configures its network interface using the information sent to it by the DHCP Server. The host now has an IP address.

The information assigned to the host by the DHCP Server has a lifetime. This is called a *lease* - for example, the host "leases" an IP address for 1 day. Before the lease expires, the DHCP Client will typically renew the lease. In this way, if a change of configuration information (such as DNS server) is needed, the network adminsitrator simply modifies the DHCP Server - the DHCP Clients in each host will retrieve the updated information from the DHCP Server.

Many computers now use DHCP to obtain an IP address, so the computer user does not need to worry about configuring their own IP address. For example, when you connect to the SIIT network with your laptop, typically you do not configure an IP address - DHCP is used.

12.2 Viewing Interface Information

By default, DHCP is used on the PCs in the Network Lab. We saw in Section 4 how to view the current network interface configuration using `ifconfig` (that is, the IP address *after* DHCP has obtained it). However the file `/etc/network/interfaces` indicates whether a dynamic (DHCP) IP address should be used, or some static (configured by the user) IP address should be used when the computer starts.

Task 19. *Look in the file `/etc/network/interfaces` to see the default configuration of your network interfaces.*

Most likely you will see a list of interfaces, with each interface (except loopback) set to use DHCP. The format of a DHCP configured interface in `/etc/network/interfaces` is:

```
auto INTERFACE
iface INTERFACE inet dhcp
```

The interface labels (eth0, eth1, eth2, ...) may vary across computers and *even* when you reboot. That is, now one network card may be referred to by eth0 and after re-booting the same card may be referred to by eth1.

To disable the use of DHCP and use static addresses, you can edit the file and change the iface section:

```
iface INTERFACE inet static
address IPADDRESS
netmask SUBNETMASK
```

12.3 Viewing DHCP Information

Now lets look at some DHCP information. The current DHCP leases are stored in /var/lib/dhcp3/dhclient.X.lease where X is the interface identifier (e.g. eth2 or eth3). Note that the lease file may contain more than one entry—the last entry is the lease currently in use.

Task 20. *View the DHCP information for interface eth2. How many days was the lease assigned for? When will the lease expire? Will DHCP try to renew the lease before it expires? Note: if there is no lease information for eth2, use eth3 instead.*

One way to refresh a leased IP address is to refresh the interface.

Task 21. *Disable and then enable an interface. Then compare the DHCP information with that in Task 20. Make note of the differences.*

12.4 Setting a Static IP Address

We may not always want to use a dynamic (DHCP assigned) IP address. In the lab, the best way to assign a static IP address is using `ifconfig`. We saw before that `ifconfig` can be used for viewing interface configuration information—it can also be used for setting interface configuration information. An example to set the IP address 10.20.30.40 (with subnet mask 255.0.0.0) to interface `eth1` is:

```
ifconfig eth1 10.20.30.40 netmask 255.0.0.0
```

You can also use `ifconfig` to enable/disable interfaces by adding `up`/`down` to the end of the command (in Linux terminology this is referred to as "bring an interface *up* or *down*"). For example, to turn off/disable/bring down the interface:

```
ifconfig eth1 down
```

And to turn on the interface (add setting a different IP address at the same time):

```
ifconfig eth1 10.20.30.41 netmask 255.0.0.0 up
```

Task 22. *Set your IP address to be a static address. Choose an IP address that is identical to that assigned previously by DHCP. Refresh the interface (down then up) and test your network connectivity after you have made the changes. Does the network connection still work? Why?*

Task 23. *Set your IP address to another value (one that is on a different network). Refresh the interface and test your network connectivity after you have made the changes. Does the network connection still work? Why?*

Task 24. *Set your network interface back so that it uses DHCP (rather than a static address).*

13 Cleaning Up

As other students (both in this lab class and other classes) must use these computers, it is very important you set the computer configuration back to the default. This includes:

- Both interfaces should be configured to use DHCP.

- Ethernet cables connected from each network card to the appropriate switch.
- All personal files deleted from the computer (e.g. source code, results).

Failure to leave your computer in an appropriate configuration may lead to a loss of marks. Leaving source code and results on the computer may result in penalties for cheating!

A Notes

Record any additional notes from this lab here (e.g. important points made by the instructor, summary of things you learned, mistakes you made). You should use this in future labs, as well as in preparation for assessment items like exams.