

Understanding Computer Networks.

Julian Oliver, June 2011

<http://julianoliver.com>

Foreword

The three texts, *Understanding Computer Networks*, *Working with the Command Line* and *Working with Computer Networks* are course notes for the **Network As Material** workshop conducted by Julian Oliver at the **Subtle Technologies Festival**, Toronto, 2011. The actual lessons were mostly performed on a customised Ubuntu Linux distribution. The examples should function however on almost any GNU/Linux distribution with the relevant software installed and many examples will also be applicable on Apple's OS X. *Note that monitor mode capture and packet injection will be a bit hit-and-miss on Apple machines.* Please email julian@julianoliver.com with any corrections, contributions, etc.

Introduction

Despite our growing dependence on Computer Networks, they remain a generally misunderstood technology.

The purpose of this document is to provide a quick introduction to the kinds of computer networks most popular today such that we can begin to better understand, traverse and even shape them. For brevity, I'll just use the word 'networks' rather than 'computer networks'.

Networks are often best understood in layers. The *OSI Layers* are perhaps the most famous description of a network. We'll work with them here.

In the workshop we worked only with layers 1, 2, 3, 4 and 7. These are arguably the most important layers:

- 1. Physical Layer**
- 2. Link Layer**
- 3. Network layer**
- 4. Transport Layer**
5. Session Layer
6. Presentation Layer
- 7. Application Layer**

The Physical Layer

This layer describes the physical infrastructure of the network; the cables, pins, plugs routers and all other devices that convey structured pulses of electricity throughout the network. In the context of wireless networking, air becomes part of the *carrier medium* in place of cable. As such the electromagnetic phenomena that constitutes wireless data packet communication can also be considered part of the physical layer.

The Link Layer

This layer describes the parts of the network responsible for moving data bits around from device to device. In order for this to happen there needs to be a sublayer that allows for hardware to be uniquely named. This is done with what is known as a *Media Access Control* address. A MAC address is a unique serial number assigned by the manufacturer.

This is a MAC address:

00:b9:a5:12:00:17

Using the Command Line tool **iwconfig** on a Linux system one can see the MAC address of the wireless access point 'FreeInternet' to which my *network interface* (wlan0) is associated. My network interface is the wireless card on my netbook:

The Access Point on a wireless network is technically known as an **ESSID**. The MAC address of the Access Point is known as the **BSSID**.

```
user@netbook:~$ iwconfig wlan0
wlan0      IEEE 802.11bgn  ESSID:"FreeInternet"
          Mode:Managed  Frequency:2.462 GHz
          Access Point: 00:b9:a5:12:00:17
          Bit Rate=54 Mb/s   Tx-Power=14 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70   Signal level=-32 dBm
          Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
          Tx excessive retries:58   Invalid misc:344   Missed beacon:0
```

The MAC address of a network interface on a Linux system can be altered using the tool **ifconfig** like so. We need to use **sudo** (*Super User Do*):to give us administrative level control.

sudo ifconfig <network interface name> hw ether <new mac address>

For instance:

```
user@netbook:~$ sudo ifconfig wlan0 hw ether 00:1a:2b:3c:4e:5f
```

Your network interface's MAC address can be used to identify you on a network. If you wish to observe the traffic to and from other users on networks I recommend changing your MAC address in advance. A tool that is somewhat more convenient to this end is **macchanger**. If it is not on your system, install it. On any Debian based Linux system (like *Ubuntu*) it can be installed like so:

```
user@netbook:~$ sudo apt-get install macchanger
```

To use macchanger you need to first take down your network interface.

sudo ifconfig <network interface name> down

We want to invoke macchanger using the **-r** switch to assign a randomly generated MAC address. So, if your network interface is **wlan0**, you would invoke it as so:

```
user@netbook:~$ sudo macchanger -r wlan0
Current MAC: 00:b9:a5:12:00:17 (unknown)
Faked MAC:   fc:2f:8d:9a:d8:10 (unknown)
```

The Network Layer

This is the layer that deals with assigning numeric addresses to actual hardware addresses. Networks need to have an addressing system separate from hardware so that packets can be routed across network boundaries and to allow for flexibility as required. The addressing system in use is the Internet Protocol (or IP). Addresses using this protocol are known as IP addresses.

The standard used for addressing currently is **IPv4**, (IP version 4). An IP address is a 32 bit address and can be expressed in any notation that accommodates this. To make it easier on people configuring networks, a dot decimal notation is used representing the 4, 8bit blocks.

Here is the IPv4 address of **http://google.de** at the time of writing:

209.85.149.99

Here is the same IP in the much less commonly used hex notation. Try it in your browser:

0xd1559563

TIP: try using this IP converter with IPs you find:

<http://www.silisoftware.com/tools/ipconverter.php>

An IP address is bound to a particular MAC address, creating a bridge between the network and network hardware in use by the host operating system. Only this way can packets make their way up to the Application Layer.

MAC address	Assigned IP Address
fc:2f:8d:9a:d8:10 →	192.168.0.17

Devices on the same physical network keep track of these assignments by sending out Address Resolution Protocol packets, or ARP packets, and updating a table on each host based on answers from the device it is querying.

IP addresses are temporary and are either *dynamically assigned* by a server on the LAN (like a *DHCP server*) or *manually assigned* using network configuration utilities on the host operating system.

More on IPv4

Each block has the range 0-255 (256 individual addresses per block).

This allows for a total of 4,294,967,296 (2^{32}) possible addresses. On Feb 3, 2011, this allocation was finally exhausted. The impending exhaustion prompted the development of **IPv6** whose 128bit *address space* allows for vastly more devices on the Internet. Regardless the transition to IPv6 is complex and very much in its infancy.

Thankfully due to *Network Address Translation* it is possible to have one device provide a gateway from an internal network of many devices (like a *Local Area Network*) to the Internet. This allows for the continued addition of *Internet facing* devices without each one of those devices having to have its own unique address on the Internet.

Within the entire *address space* of IPv4 there are three classes of address reserved for private use. The address ranges 192.168.0.0-192.168.255.255 and 10.0.0.0-10.255.255.255 and 172.16.0.0- 172.31.255.255 are addresses of this kind, reserved for use only on internal (not Internet) addressing.

Other ranges of numbers are allocated to various countries and other bodies.

The *Internet Assigned Numbers Authority* (IANA) is responsible for managing and recording the assignment of the different blocks of IP addresses. Many of the address ranges (like 187.255.255.255) are granted to regional Internet registries, responsible for assignments in their region of the world. IPs are then sold by businesses such as Internet Service Providers or delegated by government and military bodies for use with devices on the internet. Using services like GeoIP one can often isolate the country, city or even neighbourhood in which the device with the given IP is located. Many other addresses however are reserved for special uses.

Consider the *private address* 192.168.0.17. The *range* of possible devices on this network can be expressed like so, with 256 (0-255) possible devices available in the last block. The last block in the below IP is what's known as a *subnet*:

192.168.0.255

Finding the IP address of your Network Interface

Using the tool **ifconfig** on a Linux, OS X and many UNIX systems we can find the current IP of our network interface:

```
user@netbook:~$ ifconfig
```

```
wlan0      Link encap:Ethernet  HWaddr fc:2f:8d:9a:d8:10
            inet addr:192.168.1.157  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::fe2f:8dff:fe9a:d810/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:175775 errors:0 dropped:0 overruns:0 frame:0
            TX packets:144409 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:125914068 (125.9 MB)  TX bytes:22525903 (22.5 MB)
```

Note that the output of ifconfig will differ in appearance slightly across different operating systems.

Network Topologies

There are three kinds of network in popular use:

Local Area Network (or LAN)

Typically contexts include classrooms, home networks and office networks. These networks will most commonly connect to a gateway router that will allow data packets from the LAN to reach the large external network known as the Internet through a cable or DSL *modem*. Addresses in this class will have ranges of either of these:

192.168.0.0	→	192.168.255.255
10.0.0.0	→	10.255.255.255
172.16.0.0	→	172.31.255.255

Wide Area Network (or WAN)

These networks span large geographical areas and are connected by network bridges. Typical applications for a WAN might be hospital, government, industrial or military networks encompassing many buildings and/or field equipment over a large geographical area.

The Internet

The Internet is a special network comprised of many other smaller networks. These networks may be public, military, government, academic or entirely private. The entire Internet depends on the suite of TCP/IP protocols to function. Servers play an active role in the Internet, whether that *Web Servers*, *Domain Name Servers*, streaming and telephony servers (like *VoIP*), *Mail Servers*, games and file-sharing servers, to mention a few.

DNS servers have a unique and vital role as part of the Internet infrastructure. They keep an updated list of human friendly (in most cases) names known as '*domains*' like <http://hotglue.org> and <http://microsoftness.com> and the IP addresses of computers associated with them. When one types in a domain name into the URL field of their web-browser (for instance) a DNS server is queried to *resolve* that actual IP of the computer containing the service requested. When one registers a domain it is bound to an IP. Most often it is the client that has bought the domain from a *domain host* that specifies this assignment.

While the Internet has no core governing body, the [Internet Corporation for Assigned Names and Numbers](#) was established to allocate Top Level Domains like .co.nz, .hr, .ca (*name spaces*), ports and their associations with certain application types (like port 80 with HTTP), ranges of IPs (*number spaces*) to countries. It is a controversial authority as the United States, where ICANN is based, continues to have a dominant role in effecting changes made to the core aspects of the DNS system, upon which all the Internet depends.

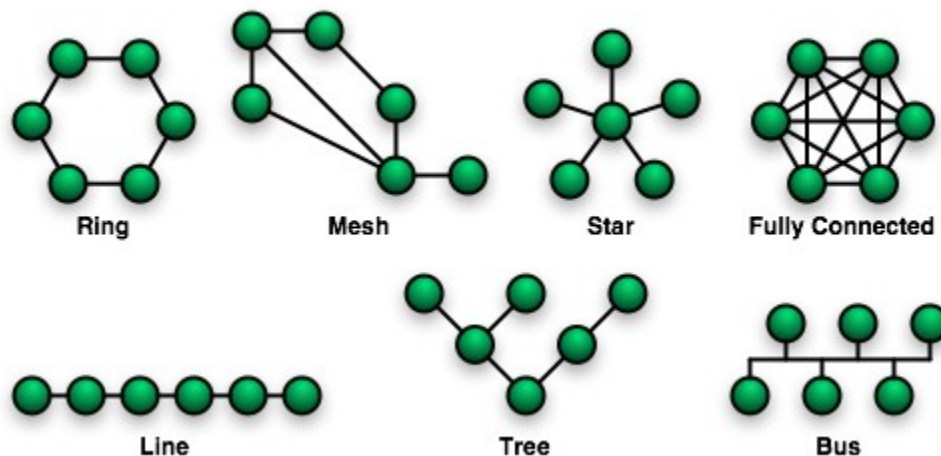
Networks and Routing

All network packets have a *source* and a *destination*. They flow between: network interfaces on the LAN, network interfaces on the LAN and the Internet (typically on *servers*) and between network interfaces on the Internet. The flow of packets on a network is described as a *route*.

Devices responsible for routing packets are *routers*. Routers are also vital in that they provide a means for packets to pass from one network to another.

Part of a router's job is to read network packets looking for the ultimate intended destination of that packet. If it is reachable by the router is then *forwarded* it on its way. Packets intended for networks or addresses not reachable by the router may be *dropped*.

Ultimately it is the route of packets that defines the Topology of the network. Here are some typical topologies:



The Topology most commonly used on LANs is the *Star*.

In *SoHo* (Small Office Home Office) configurations that offer wireless connectivity, the central node of the star is quite often a single device that combines the *Wireless Access Point*, *Gateway Router* to the Internet (via a service like *DSL*) and the *Local Area Network Router*.

On networks that use cable for their Internet connection, a separate *Cable Modem* often acts as the *Gateway*. To this device a wireless router is attached, providing Internet access to LAN clients.

The Transport Layer

This layer is represented by several protocols that define end-to-end, packet-based communication between applications. Every network packet has a *header* and a *body*. The header includes information such as the source and destination of the packet, the delivery protocol in use, the length of the packet and other information. The body contains the data intended for display in an application (see *Application Layer* below).

There are two primary protocols in popular use, the *Transmission Control Protocol* over IP (known as TCP/IP) and the User Datagram Protocol (or UDP).

TCP/IP prioritises reliable end-to-end packet delivery over speed of delivery. It is a connection-based protocol that orders out-of-order packets, requests retransmission of lost packets, manages packet duplication and performs error checking throughout the entire

network conversation. Typical services using TCP/IP are email, FTP, SSH and P2P file sharing services.

UDP however provides a fast yet unreliable packet delivery service. With none of the error-checking overheads of TCP, it is a network efficient protocol well suited to applications like streaming media, network gaming and VoIP.

Network ports can be thought of as doors on a building. Some are open and some are closed. Ports are opened and closed by applications as required. Ports typically define sub protocols like SMTP (port 25, sending mail), SSH (port 22, secure shell connections) and HTTP (port 80, transport of web pages). Firewalls can restrict connections made to certain ports (from all or certain addresses) while privileging others.

The Application Layer

The Application Layer is comprised of the applications that are ultimately responsible for requesting packets and displaying their contents. The most commonly used application in this layer is the web browser. Web browsers use the *Hyper Text Transport Protocol* or HTTP. HTTP use port 80.

When working with network traffic it is possible to filter and route that traffic based on the ports that are used. Firewalls filter based on both ports and network addresses.

Further reading

A good next step would be learning about the structure of a Network Packet. Here the Wikipedia page is as good as any, though you can also just start up **Wireshark** and explore packets of different types in the GUI itself. If you come across an acronym or term you don't understand, search for it online. Thankfully it's a thoroughly covered area.

There are a great deal of free books on Computer Networking out there on the Web. I can recommend looking through this list:

<http://freecomputerbooks.com/unixNetworkBooks.html>

If you are willing to spend money, Gary A. Donahue's *Network Warrior* and Bruce Hartpence's *Packet Guide to Core Protocols* will be invaluable purchases.

Next, see ***Working with the Command Line***.