

## Introduction to MS Windows Network Programming

### Aims:

The aim of this worksheet is to introduce tools that are useful in determining the nature of a network, e.g. **ipconfig**, **ping** and **arp** and local processes, **taskmgr**. It also takes a first step into using 'well known' ports by experimenting with standard *client/server* facilities, e.g. **telnet** and **echo** and examines the files that must be maintained for network access. The assumption is that the information is needed from within a **Command Prompt** terminal.

### Tools.

**ipconfig**. Internet Protocol Configuration.

Using `ipconfig /all` gives a list of all of the network interfaces installed and configured on the host machine. In an appropriate command prompt window type:

```
ipconfig /all
```

This should display details of the loopback interface. This interface is useful if we have no connection to a network or to test programs before using the network itself. Details of the Ethernet interface (`eth0`) details should be something like:

Windows IP Configuration

```
Host Name . . . . . : BUGLE
Primary Dns Suffix . . . . . : cems.uwe.ac.uk
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : cems.uwe.ac.uk
                                   uwe.ac.uk
                                   netlab.cems.uwe.ac.uk
```

Ethernet adapter Local Area Connection 10:

```
Connection-specific DNS Suffix  : cems.uwe.ac.uk
Description . . . . . : Intel(R) PRO/100 VE Network Connection
Physical Address. . . . . : 00-04-23-2C-79-CA
Dhcp Enabled. . . . . : No
IP Address. . . . . : 164.11.243.243
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 164.11.243.1
DNS Servers . . . . . : 164.11.8.21
                                   164.11.8.4
```

It should be easy enough to guess what the fields mean. From the details for `eth0`, write down the IP address of the host machine (these could be useful later!). To find out more information on **ipconfig** try using:

```
ipconfig /?
```

**netstat. Network Statistics.**

Another way to obtain information about the network interfaces is to use **netstat**. Not only can details of interfaces be displayed but information about any servers that are running.

Try:

```
netstat -ao
```

This gives protocols of current TCP/IP connections and the PID of the associated process.

Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	BUGLE:http	BUGLE.cems.uwe.ac.uk:0	LISTENING	1476
TCP	BUGLE:epmap	BUGLE.cems.uwe.ac.uk:0	LISTENING	980
TCP	BUGLE:https	BUGLE.cems.uwe.ac.uk:0	LISTENING	1476
TCP	BUGLE:microsoft-ds	BUGLE.cems.uwe.ac.uk:0	LISTENING	4
TCP	BUGLE:3306	BUGLE.cems.uwe.ac.uk:0	LISTENING	1840
TCP	BUGLE:3389	BUGLE.cems.uwe.ac.uk:0	LISTENING	904
TCP	BUGLE:microsoft-ds	localhost:1131	ESTABLISHED	4
TCP	BUGLE:1050	BUGLE.cems.uwe.ac.uk:0	LISTENING	2684
TCP	BUGLE:1131	localhost:microsoft-ds	ESTABLISHED	4
TCP	BUGLE:netbios-ssn	BUGLE.cems.uwe.ac.uk:0	LISTENING	4
TCP	BUGLE:1122	nas.cems.uwe.ac.uk:microsoft-ds	ESTABLISHED	4
TCP	BUGLE:1245	dgen-uwe04.campus.ads.uwe.ac.uk:1025	ESTABLISHED	2620
TCP	BUGLE:1246	dgen-uwe04.campus.ads.uwe.ac.uk:1025	ESTABLISHED	2620
TCP	BUGLE:1248	EGEN-UWE02.uwe.ac.uk:1182	ESTABLISHED	2620
TCP	BUGLE:1249	dgen-uwe04.campus.ads.uwe.ac.uk:1025	ESTABLISHED	2620
TCP	BUGLE:1250	dgen-uwe04.campus.ads.uwe.ac.uk:1025	ESTABLISHED	2620
TCP	BUGLE:1258	skarloe.y.cems.uwe.ac.uk:microsoft-ds	ESTABLISHED	4
TCP	BUGLE:1269	emily.cems.uwe.ac.uk:microsoft-ds	ESTABLISHED	4
UDP	BUGLE:microsoft-ds	*:*		4
UDP	BUGLE:isakmp	*:*		740
UDP	BUGLE:1025	*:*		1108
UDP	BUGLE:1026	*:*		1108
UDP	BUGLE:4500	*:*		740
UDP	BUGLE:6004	*:*		2620
UDP	BUGLE:ntp	*:*		1068
UDP	BUGLE:1027	*:*		740
UDP	BUGLE:1045	*:*		684
UDP	BUGLE:1057	*:*		1628
UDP	BUGLE:1900	*:*		1232
UDP	BUGLE:ntp	*:*		1068
UDP	BUGLE:netbios-ns	*:*		4
UDP	BUGLE:netbios-dgm	*:*		4
UDP	BUGLE:1900	*:*		1232

Notice that the column labelled '*Local Address*' gives `hostname:portNumber`.

Getting all the **netstat** details may give rather a lot of information, so it might be sensible to pipe the output into **find** to detect just those bits needed, e.g.

to list those entries with the word Microsoft in them:

```
netstat -ao | find "microsoft"
```

to list entries using port number 3456:

```
netstat -ao | find "3456"
```

This tool can be used to get the name of the associated process. Try:

```
netstat -abo
```

This is often slow in getting all the data and gives rather a lot of it, so it might be best to pipe it into **more** so that the output is paginated or into **find** to locate specific lines, e.g. type:

```
netstat -abo | more or netstat -abo | find "day"
```

and notice the difference. Note whether the daytime and echo servers are running, we shall need these for testing some of our socket programs. If they are not running we need to get them started (this is probably not possible unless you have administrator privileges!).

### **ping.** Packet Internet Groper.

This tool may be used to discover whether a particular host is connected to the network. It repeatedly sends packets of data to the requested host and expects the data to be echoed back. Try **pinging** the loopback address by:

```
ping 127.0.0.1 or ping localhost
```

Try **pinging** the IP address of the neighbouring machine.

Note that if we `ping localhost` the display actually shows the IP address of *localhost*. Determine the IP address of *kenny* (the 'controller' of our little network!). Something to think about - How does the operating system resolve the host names to IP addresses? (see later!)

### **hostname.** Get Host Name.

To get the name of the machine we are working on type:

```
hostname
```

### **arp.** Address Resolution Protocol.

Used to display or set IP address MAC address pairs, i.e. associates IP addresses to their hardware addresses. Try:

```
arp -a
```

For more information on these tools use Windows Help, surf the Web or use the `/?` or `-?` Switch after the command.

### **taskmgr.** Task Manager.

We often need to check whether our processes are running. This can be achieved by using the **taskmgr** from a window. Type:

```
taskmgr
```

This should produce the standard Task Manager pop-up that we often use in Windows, as a result of Ctrl+Alt+Del, to kill hanging processes. For our purposes it is convenient to see the

**Process Identifier (PID)**, so select the Processes tab then View -> Select Columns and tick PID and any other fields you think necessary!

It should be possible to relate information displayed by **netstat** to this display from **taskmgr**.

### 'Well known' Ports.

Standard services are connected to 'well known' ports. These ports have servers actively listening on them at all times (if they have been activated!). Clients can connect to these services to obtain the 'data' they supply. A useful *client* for us to use to test our own network servers is **telnet**. Useful *servers* for us to test our own clients are the daytime service and the echo service. When a connection is established either the client or the server may close the connection (perform an *active close*!). Try getting the date and time of day from the *localhost* by:

```
telnet localhost daytime
```

Try replacing `localhost` with the IP address of your host, your neighbour's and perhaps kenny. Which end performed the *active close*?

Now try connecting to the echo server. You will need to type in some text to be echoed and close the connection using Ctrl-]. Finally end the **telnet** session with *quit*.

### Files used in Network Access.

#### **C:\windows\system32\drivers\etc\services.**

This lists all the possible network services, their port number and the protocol they use, even if the server is not actually running. Try typing:

```
more C:\windows\system32\drivers\etc\services
```

What are the 'well known' port numbers for daytime and echo? Try to **telnet** to these services by their port number rather than their name.

If we were to create a new service then we would need to add details of that service to this file. Unfortunately we need to have System *Administrator* access to do this!

#### **C:\windows\system32\drivers\etc\protocol.**

Just as services are referred to by a number so are protocols. The file **protocol** lists the names of the protocols together with their associated number and the equivalent name for that number. If we were to invent a new protocol we would add to this file! Inspect the file using *more*. What are the numbers associated with tcp and udp?

#### **C:\windows\system32\drivers\etc\hosts.**

A local list of host names and IP addresses. This is used to resolve host names to IP addresses for use by tools such as **ping**. As *administrator* we can alter this list to hold details of hosts we contact often. It remains static, the details entered by *administrator* do not change dynamically as we access the network.