Computer Networks - Lab Session 1
Part I - Basic network configuration

February 2013

1 Introduction

1.1 Networks

You undoubtedly remember that a network is an interconnection of several workstations in order to carry information from one host to another. The network that you will have to set-up will be of *Ethernet type*.

1.2 Ethernet address and Internet name

Every network card has an Ethernet address, which identifies it and is assigned to it permanently during the manufacturing process (no two cards can have the same Ethernet address). Such an address consists of six bytes. The usual notation for these Ethernet addresses is to state the six hexadecimal bytes and separate them with a colon `:`. For example: `8:0:20:4:69:d6`

Note : this address is set (generally) when the hardware is manufactured.

Given that not everyone necessarily uses Ethernet networks, other addressing systems can be encountered, and networks of different types may be interconnected, so it is necessary to assign a logical address to each host that allows one to identify the nature of the network. In our case, we will be using **Internet addressing**.

Such an address is composed of:

a network address : a number that identifies the network on which the host is located;

a host address on the network: a number that identifies the host within this network.

This address consists of four bytes, and is expressed in the form `n1.n2.n3.n4`, in which `ni` is the decimal value of a byte. Also note that such an address can be chosen by the network administrator.

Let remember that some IP ranges (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) are reserved for private use in order to isolate the different IP networks and to guarantee the uniqueness of an IP subnetwork. You should choose your addresses in one of those ranges.

2

2.1 Setting-up the network

The first stage of the practical exercise to be performed on your network is to check that it works properly.

2.1.1 Hardware connection

Operation 1

Physically connect the four workstations to the same network (cf. fig. 1). Only use the `bge0` interface of each system.

2.1.2 IP configuration

Your network is ready. Now you have to configure the stations at software level, so that they recognize each other and so that they can interact. You can make your network operational in two different manners:

- Either you edit the configuration files and then restart the systems (or simply run the script `/etc/netstart`) so that they put your modifications into effect;
- Or you can manually run commands that enable you to instantly configure your systems.

> *In the real life, an host often acquires dynamically its IP configuration from a server. What protocol offers this functionality?*

Choice of the Internet addresses

Operation 2

> *Choose a range of addresses and justify this choice.*
> *From within this range, choose an address for your network.*
> *Lastly, choose an address for each station*
>
> *In the diagram in Figure 1, summarize the various addresses selected. Give reasons for the choices you made in the above instructions.*

Manual configuration of workstations

The command used for configuring Ethernet interfaces is `ifconfig` (InterFace CONFIGuration). Configuring an interface involves initializing it, assigning it a certain number of information items (the hardware system's Internet address, for example) and, lastly, declaring it *operational.*

Operation 3

Before configuring the interface of a hardware system, you need to give it a name. Interface names always take the form: `<type><number>` (for instance: `bge0`, or `fxp1`). Here, we are only using the second interface of each hardware system; it is called `bge0`. You can look at the `ifconfig` manpage (`man ifconfig`) and also read the interface driver specifics (`man em`)

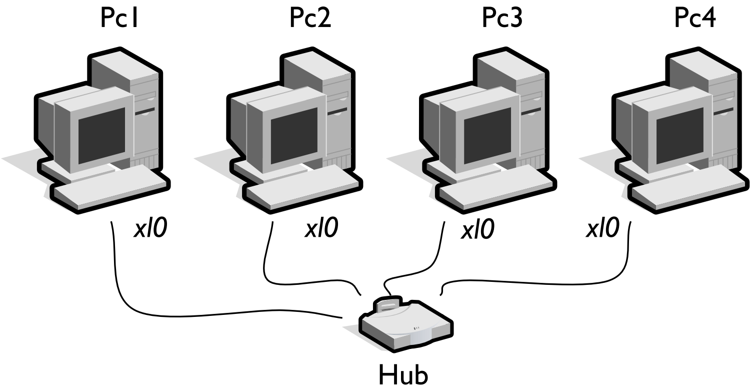

Figure 1: Network diagram

Interface configuration : the purpose of the interface is to connect the hardware system to the network. Therefore, you should assign it a number of information items, notably the station's Internet address on the network.

The configuration command using `ifconfig` is as follows:

```
# ifconfig <interface_name> <internet_address>/<prefix length> [up]
```

Configure the interface of the four workstations in accordance with the information stated above. The hardware system will now be able to interact via the Ethernet cable.

Note: in this case, you don't need to reboot the systems for your network to be operational, but they will lose their configuration if you turn them off, and you will have to re-execute the same commands when you switch them on again.

Checking the status of interfaces : you can check the status of your interfaces at any time. To ascertain the status of the interface you have just configured, enter the following command:

```
# ifconfig [<interface_name>]
```

You will obtain output similar to the following:

```
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        inet6 fe80::2d0:b7ff:fe4d:d519%bge0 prefixlen 64 scopeid 0x1
        ether 00:d0:b7:4d:d5:19
        media: Ethernet autoselect (none)
        status: no carrier
```

The information that might be of interest within the output from the command might be: whether or not the interface is UP; the Internet addresses, IPv4 and IPv6, the Ethernet address of the device (`ether`).

To restore the interface to initial status (so that it can be reconfigured into another configuration), the command is as follows:

```
# ifconfig <interface_name> delete
```

`# ifconfig <interface_name> down` enables you to disable an interface without changing its configuration. This is what is done with the bge0interface.

Configuration by editing the configuration files

In real life, you don't configure systems by hand (except in exceptional situations): you edit the configuration files that hardware systems read automatically when they boot. When they read these files, they perform the same operations that you have just executed.

In this section, you will edit the configuration files. All the configuration files that you will edit are to be found in the /etc directory of the systems.

All the configuration files are ASCII files that you can edit with your preferred text editor. In these files, the hashmarks (#) mark the beginning of a comment.

Even if the system manages to detect the presence of Ethernet interfaces, it does not yet configure them. To tell it that an interface should be configured, you have to edit the file `/etc/rc.conf`. For example, the string that follows "`ifconfig_bge0`" is given as arguments to the command "`ifconfig bge0`" when the system is initialized. You should take a look to `man rc.conf` to obtain the documentation on this function.

‖ *In particular, what is the purpose of `gateway_enable="yes"`?*

Every IP address can be stated in decimal format, but for convenience reasons it is possible to assign it a symbolic name (this address will feature in several configuration files, and in many commands. This association must be specified in the file /etc/hosts. The symbolic name will, in fact, be the official name of each hardware system (in this case, pc1, pc2, etc.).

The lines in the file /etc/hosts are in the following form:

```
<Internet address> <official name of the system>
```

Operation 4

‖ *Create the file `/etc/hosts` on each system, and enter their content.*

This file is read every time a name resolution takes place, as in the following section. There is nothing to do for the system to take it into account, whereas *e.g.* `/etc/rc.conf` is read upon system startup or when the script `/etc/netstart` is launched manually.

2.1.3 Checking the network

Using `ping`

Now you need to check that the systems are indeed properly connected and properly configured. There is a standard tool for this: ping. By default, ping lets you check that a remote system does indeed respond when contacted.

Operation 5 On one station, enter the following command:

```
# ping <address_of_remote_system>
```

If you don't receive a response to the ping, informing you that the remote system is up and running, it means that there is a problem. If you don't get a response, check your setup and repeat a ping.

Using Internet addresses in decimal notation at user level is not flexible. It is more convenient to use a name instead (pc1, pc2, etc.). To this end, you need to have entered all the address/name associations in the hosts file of each system.

> *Make the appropriate changes (if needed) to these* `/etc/hosts` *files and repeat the above operation, using the official names.*

Procedure for logging-in to a remote system

You can use ssh for remote login.

Operation 6 Enter :

```
# ssh <user_name>@<workstation_name>
```

At the password prompt, enter the password associated with the user name specified in the command. Once your log-in has been accepted, you can work on the remote system in the same way as you can on a local system.

Note you can see that the configuration of a system as regards network connections is very simple (`ifconfig`). After these few operations, you can set-up sessions (`telnet, rlogin`, etc.), and perform file transfers (FTP,SCP). In reality, you should understand that the use of such tools for communications requires the presence of special processes on each system. For more information, refer to man inetd.

2.2 Observation of network activity

Reminder : to communicate, systems interchange information items in the form of packets, which are the unit of data exchanged over the network.

After having configured the systems and checked that, at user level, everything is working OK, then you can now listen to the Ethernet cable and watch what happens when you run commands such as ping.

The tool that allows you to observe the network is called tcpdump. You can also use the graphic tool wireshark, which lets you spy on network traffic as with tcpdump, but gives you a more easily-readable display.

2.2.1 Observing the `ping` command

On one of the four hardware systems, run tcpdump with the argument -i followed by the name of the interface :

Operation 7

```
# tcpdump -i bge0
```

tcpdump then displays a message telling you which interface it will listen-in to and in what manner: by default, it listens in promiscuous mode, i.e. it sniffs and analyzes all the frames on the network ? even those that don't concern the system on which it is running. Then tcpdump waits for something to happen on the network. On a second system, execute a ping to see whether a third system is running:

```
# ping <workstation_name>
```

> *Remembering that ping uses packets of ICMP type, analyze and comment on its operation.*

In addition to packets that directly concern ping (ICMP), tcpdump should sniff other packets of ARP type (Address Resolution Protocol) if this is the first time you attempt to reach a system (or if some time has passed). If this is not the case, enter the command arp -a -d on the system from which the ping was executed, and then run another ping.

> *Analyze and interpret the ARP packets.*

> *Examine the analysis from tcpdump: packets concerning ARP protocol only feature at the beginning. Why?*

Operation 8

On a third system, start `wireshark`. When two packets have been sniffed, stop wireshark and then look in detail at the contents of the packets.

Locate the values of the various fields of the packets in hexadecimal format (ETHER header and ARP data).

> *What lets you identify the packets as being of ARP type?*

Given that the Ethernet frames take the following form:

offset →	0	6	12	14	60–1514	64–1518
Preamble *(64)*	Destination @ *(48)*	Source @ *(48)*	Type *(16)*	Data *(368-12000)*	CRC *(32)*	*(in bits)*

> *Where is the ARP packet in this frame?*

> *How does the Ethernet level ascertain the size of the packets that it receives? How does it establish where the end of the packet is located?*

> *What is sent to the layer above (3)?*