# Lab 1 Answers

## Tasks 1 & 2

An example of running “ifconfig” can be seen below. I only provide a small snippet on the output since on machines with multiple interfaces it can be a quite long and messy output.

```
(base) Angeloss-iMac:~ akm$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=50b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV,CHANNEL_IO>
        ether 3c:cd:36:66:59:b9
        inet6 fe80::425:3fbe:dd93:ca64%en0 prefixlen 64 secured scopeid 0x4
        inet 191.167.1.110 netmask 0xffffff00 broadcast 191.167.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex,flow-control,energy-efficient-ethernet>)
        status: active
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 3c:22:fb:58:9c:aa
        inet6 fe80::47b:a73d:6d8e:f977%en1 prefixlen 64 secured scopeid 0x8
        inet 168.254.193.66 netmask 0xffff0000 broadcast 168.254.255.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (<unknown type>)
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:35:ea:26:30:00
        media: autoselect <full-duplex>
        status: inactive
```

As evidenced, there are multiple network interfaces on my machine

- lo0 -loopback interface,
- gif0 – generic software interface for Apple Wireless Direct link,
- stf0 is the IPv4/IPv6 tunnel interface for supporting IPv4 to IPv6 transition
- en0 – ethernet 0
- en1 – WiFi interface
- en2 – Apple Thunderbolt interface

We can see that I am connected with a wire and I have an IP address (**191.167.1.110**), a broadcast address (**191.167.1.255**) and my ethernet card’s Mac address is : **3c:cd:36:66:59:b9.**

If I also run “networksetup -listallhardwareports” I will get a leaner view highlighting the MAC addresses for my network interfaces having (I copy only for my en0):

```
Hardware Port: Ethernet
Device: en0
Ethernet Address: 3c:cd:36:66:59:b9
```

For this lab, we focus on the MAC address.

## Tasks 3 & 4

The output for “arp -a” is provided below:

**angeloss-iphone.home (191.167.1.66) at a8:8e:24:1:3f:b4 on en0 ifscope [ethernet]**
**iphone.home (191.167.1.105) at 9e:c0:3a:85:cf:49 on en0 ifscope [ethernet]**
**angeloss-mbp-2.home (191.167.1.108) at (incomplete) on en0 ifscope [ethernet]**
**angeloss-imac.home (191.167.1.110) at 3c:cd:36:66:59:b9 on en0 ifscope permanent [ethernet]**
**unknown-cc-4e-ec-21-26-ac.home (191.167.1.128) at cc:4e:ec:21:26:ac on en0 ifscope [ethernet]**
**bthomehub.home (191.167.1.254) at 40:f2:1:2c:89:a2 on en0 ifscope [ethernet]**

As shown, this lists the devices running in my local network with their corresponding domain names (e.g., angeloss-iphone.home), IP, MAC addresses and from which interface there are visible. Hence, all devices are seen/routed to/from my machine through the en0 interface (i.e. the Ethernet interface, thus from the ethernet network in my home).
Some peculiarity exists on some of the statements in some of the entries. For instance there is an incomplete connection to one of my devices (angeloss-mbp-2.home). In this scenario my machine has initiated an ARP request through my router (bthomehub.home) over the Ethernet network and to that particular device, with no response yet therefore there is no MAC address in the entry. Also, the entry for my actual iMac device (angeloss-imac.home) as seen over my ARP cache is denoted as permanent (i.e. static) in contrast to the rest. As expected, the rest of the devices/hosts are not listed as permanent entries, thus they are considered as dynamic and they were dynamically discovered through ARP when I initiated “arp -a”. Another observation is that the all MAC addresses do not share the same 24-bit signature (i.e. the first 3 octets in each address – e.g., cc:4e:ec) implying that all these interfaces come from different vendors.

Looking at the OUI lookup and also on whatsmyip.org I can find the following for all the devices

**angeloss-iphone.home, a8:8e:24 , Apple Inc**
**iphone.home, 9e:c0:3a, unknown, no MAC address available**
**angeloss-imac.home, 3c:cd:36, Apple Inc**
**unknown-cc-4e-ec-21-26-ac.home, cc:4e:ec, Humax Co. LTD**
**bthomehub.home, 40:f2:1, unknown**

Indicating that despite standardisation on MAC addresses, you cannot always find the vendor due to the massive production of devices by many vendors. Nonetheless, using the ARP cache in your local network can always help you on getting some general diagnostics.

## Task 5

The standard way to “force” an entry to be added to the ARP cache (other than through acquiring administrative rights and adding the entry by hand), is by simply initiating some communication with the target host. This can be done via such commands as `ping`, or even by merely entering the target IP as a web address in a browser window; in the latter case no web page will (most probably) be displayed, but the data-link (and network and transport) layer(s) will have been engaged in the process. If the target computer is on the same link as ours, there should then be an entry added to our ARP cache. Once this is done, we can use `arp -a` again to retrieve the new entry.

## Task 6

As discussed previously, by merely accessing the web page itself, an entry will be added to the ARP cache of our computer. However, in this case you will not be able to get a local ARP cache entry of the web server since you are not in the same network. If you were in the SoCS labs you would be able to get an entry.

## Task 7

For this task, we can access the web server of the University just fine. However, no relevant entry is added to the ARP cache. This is typically due to the fact that the target host is not on the same link as our computer; that is, data travelling from our computer to the web server and back needs to go through one or more routers. Unless said routers are configured to relay ARP traffic, we won't be able to retrieve the MAC address of the web server. This isn't a problem as the end-to-end communication is handled by protocols at higher layers of the OSI model.