Wireless LANs Labs

1 Wireless Lab Specification (C6)

1.1 Introduction

The wireless lab in C6 is isolated from the main university network, and allows for the development of mobile networks and applications, for both projects and teaching. It currently contains the following:

- **20 wireless hosts with Belkin IEEE 802.11b/g wireless cards**. Note: Do **not** set the wireless cards to have an address which links to the Ethernet network (192.168.1.*x*). The Ethernet network is used to allow the connection to the Aironets, and the wireless network should have addresses which do not link to the Ethernet network.
- **12 wireless hosts with Cisco Aironet IEEE 802.11g wireless cards**.
- **One Cisco 3560 switch** (C6SW2).
- **Seven Aironet 1200 wireless access points**.
- **One Windows 2003 server**. This server has two Ethernet cards, which allows it to be part of the main Ethernet network (192.168.1.5), and also the Wireless network (such as 192.168.2.*x*). The main connection allows it to be configured to be part of the wireless network.
- **One Linux server**. This server has two Ethernet cards, which allows it to be part of the main Ethernet network (192.168.1.6), and also the Wireless network (such as 192.168.2.*x*).

The main Ethernet network is located on the 192.168.1.*x* network, where the main server is at 192.168.1.1, and the hosts start at 192.168.1.6 (on the left-hand side of Bench 1) and go onto 192.168.1.25 (on right-hand side of Bench 4), as illustrated in Figure 1.

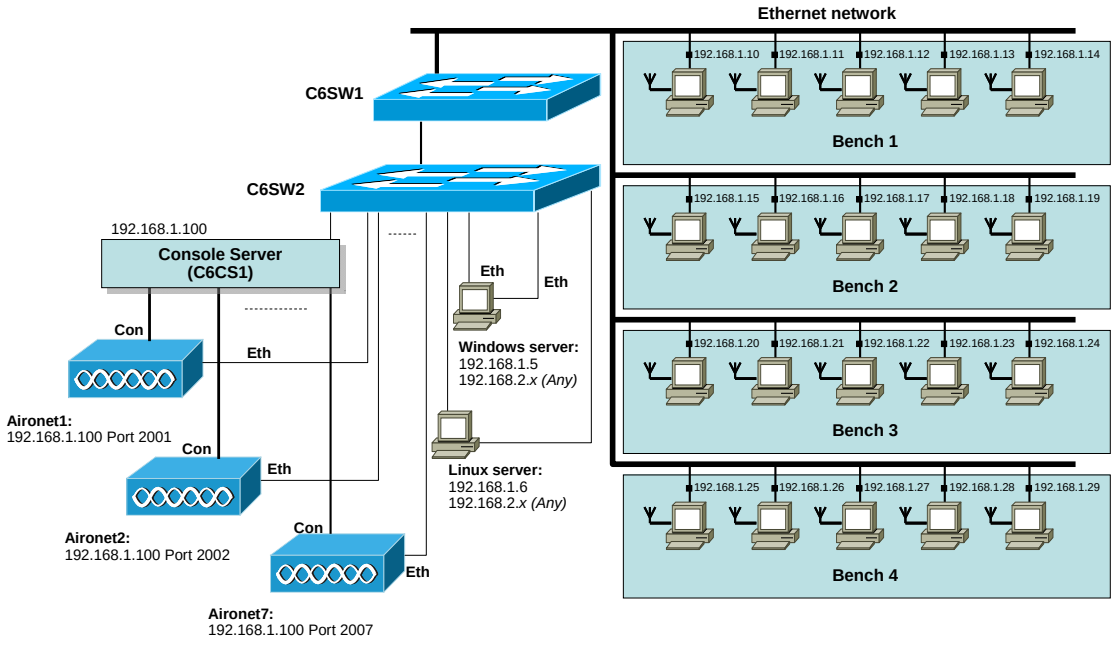


Figure 1: Outline of wireless lab setup

1.2 Example configuration

The following configuration sets up:

	Tutorial example	Your example
Device:	[Aironet1]	[ ]
Remote port:	[2001]	[ ]
Aironet1 IP:	[192.168.2.3]	[ ]
Aironet SSID:	[bill]	[ ]
Wireless client:	[192.168.2.2]	[ ]
Windows server:	[192.168.2.5]	[ ]

Figure 2 illustrates the example setup. Please note that your connection is likely to be **different**, as you want to have different IP addresses and SSIDs to other wireless networks.

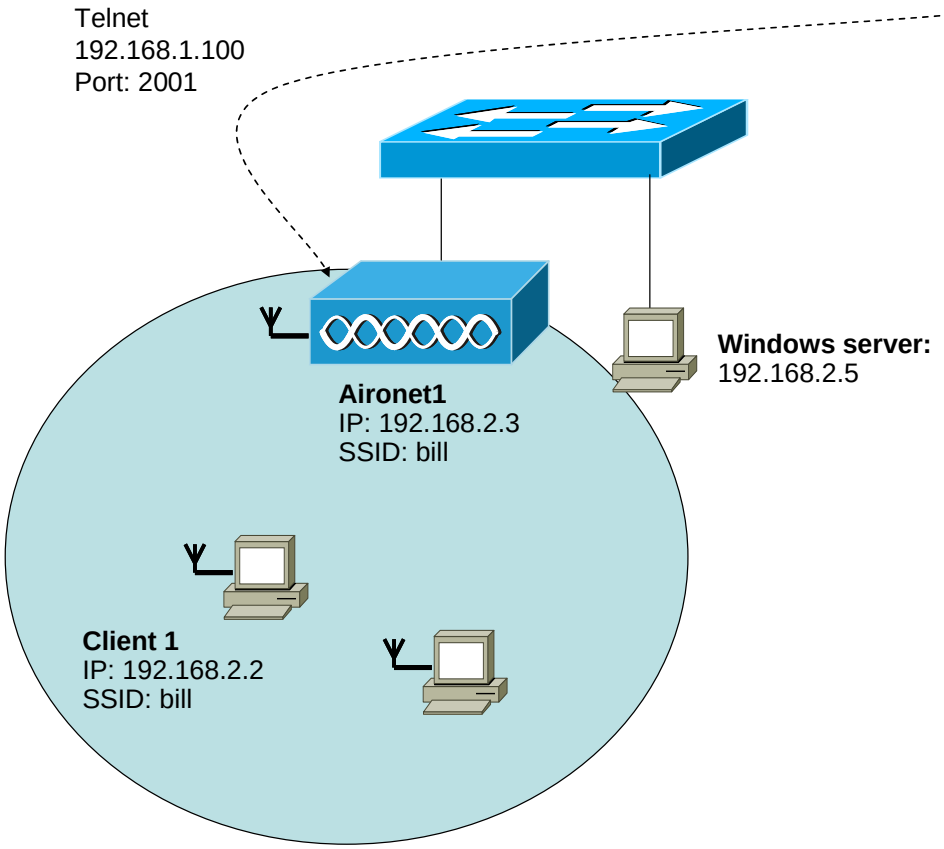


Figure 2: Example setup

1.2.1 Connection to Aironets

There are currently seven Cisco Aironet 1200 wireless access points, which can be configured by connecting to the console port of the Aironet, and using a Telnet connection (Figure 3). These are accessed by:

Aironet1: Telnet address: **192.168.1.100** Telnet port: **2001**
Aironet2: Telnet address: **192.168.1.100** Telnet port: **2002**

Aironet3: Telnet address: **192.168.1.100** Telnet port: **2003**
Aironet4: Telnet address: **192.168.1.100** Telnet port: **2004**
Aironet5: Telnet address: **192.168.1.100** Telnet port: **2005**
Aironet6: Telnet address: **192.168.1.100** Telnet port: **2006**
Aironet7: Telnet address: **192.168.1.100** Telnet port: **2007**

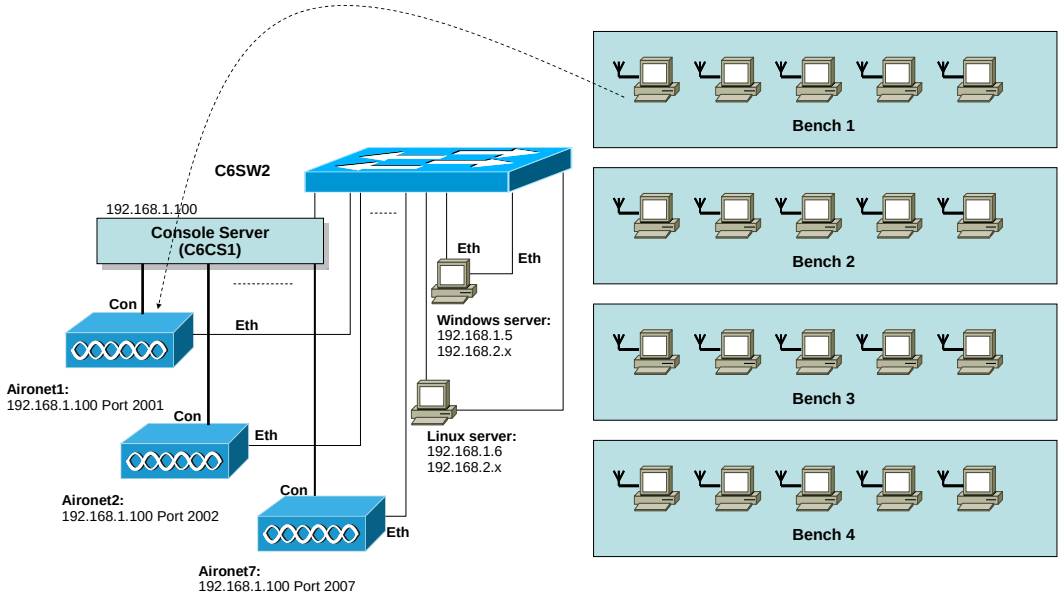


Figure 3: Outline of connecting to the Aironet console ports

Figure 4 shows an example connection to [Aironet1], where the remote port is [2001]. The connection in this example uses Windows HyperTerminal (Start->Programs->Accessories->Communications->HyperTerminal). The connection to the Aironet should then be made (such as shown in Figure 5). With this, the password should be **Cisco**, after which, the main login to the Aironet is made (Figure 6), which also has a password is **Cisco**.

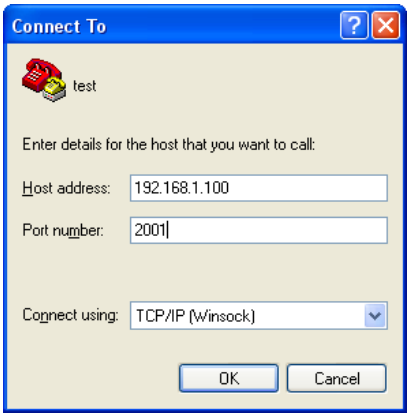


Figure 4: Telnet connection to the Aironet (Aironet1)

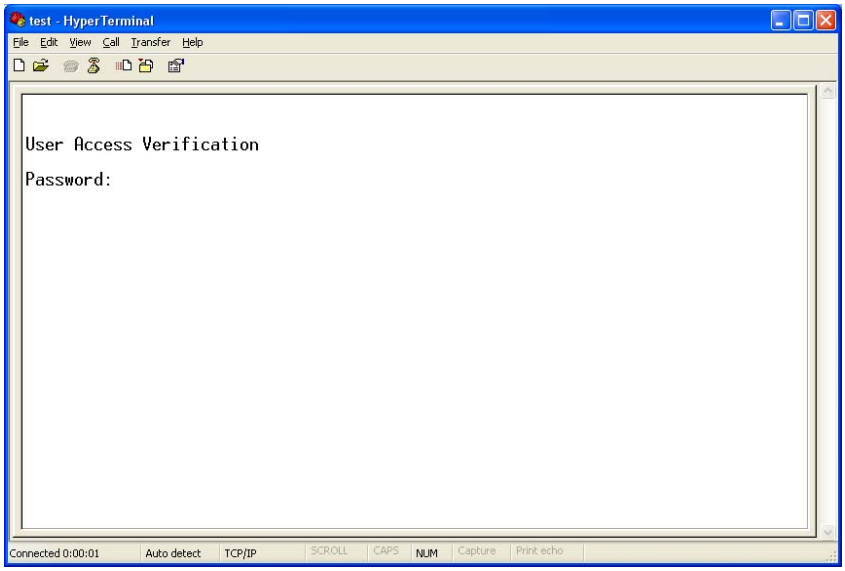


Figure 5: Telnet connection to the Aironet (Aironet1)

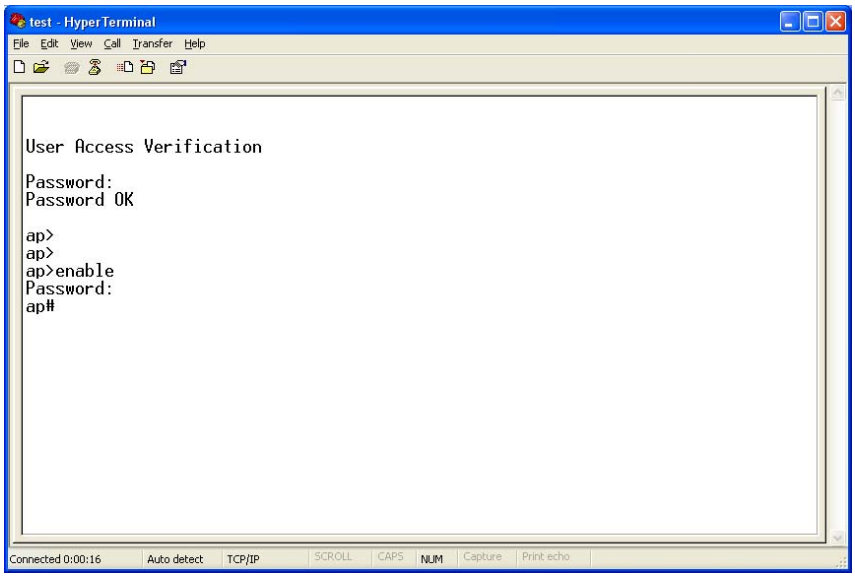


Figure 6: Telnet connection to the Aironet (Aironet1)

An example setup is shown in Figure 7. The configuration is this case is:

```
> enable
# config t
(config) # hostname ap
(config) # dot11 ssid bill
(config-ssid) # authentication open
(config-ssid) # exit
(config) # int bvi1
(config-if) # ip address 192.168.2.3 255.255.255.0
(config-if) # exit
(config) # int d0
(config-if) # channel 6
(config-if) # ssid bill
(config-if) # no shutdown
(config-if) # exit
(config) # int fa0
(config-if) # no shutdown
(config-if) # exit
```

This sets up the IP address of the access point at [192.168.2.3] with an SSID of [Bill] and using radio channel 6. The wireless nodes which connect to this access point will now have an address of 192.168.2.*x*.

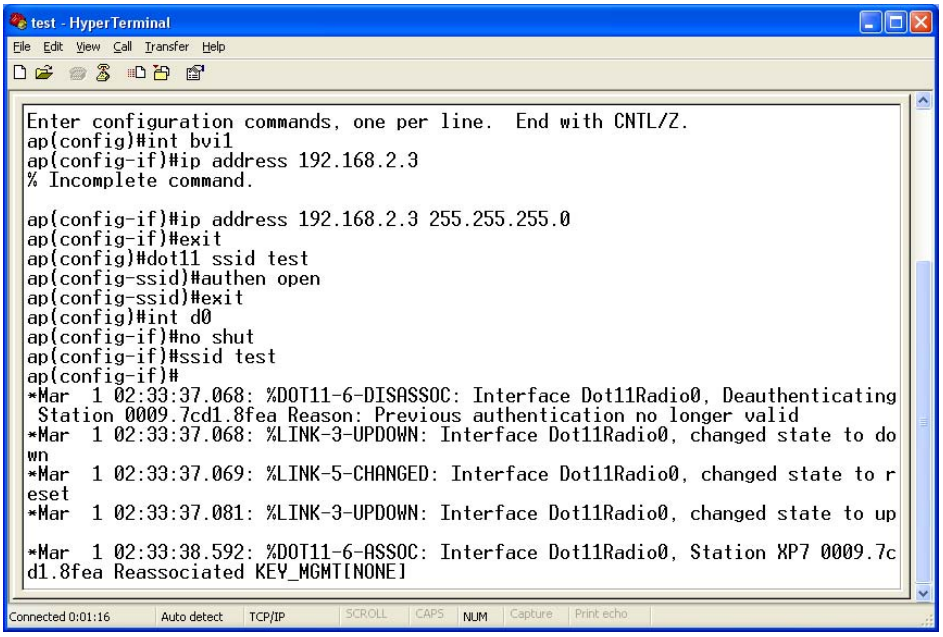


Figure 7: Telnet connection to the Aironet (Aironet1)

1.2.2 Setting up the Wireless client (Cisco 350)

Each of the hosts has a wireless card, such as a Belkin client (Appendix 1) or a Cisco 350 card. As an example the following sets up a connection to the [192.168.2.x] network. Initially the Cisco Client program is used to setup a profile (Figure 8/9), after which the SSID is set to the setup on the access point (Section 2), as shown in Figure 10, which, in this case, is [Bill].

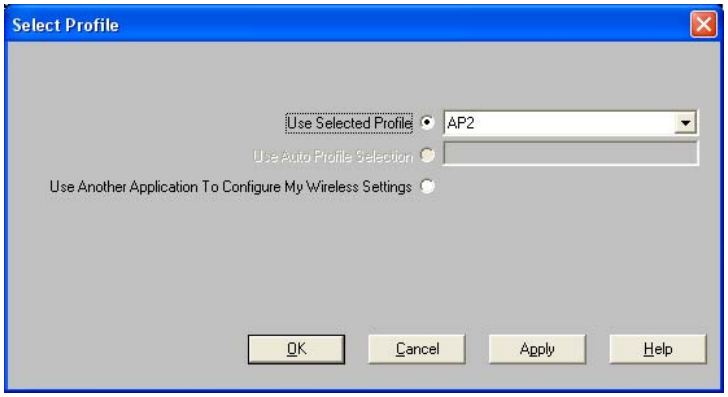


Figure 8: Selecting a profile

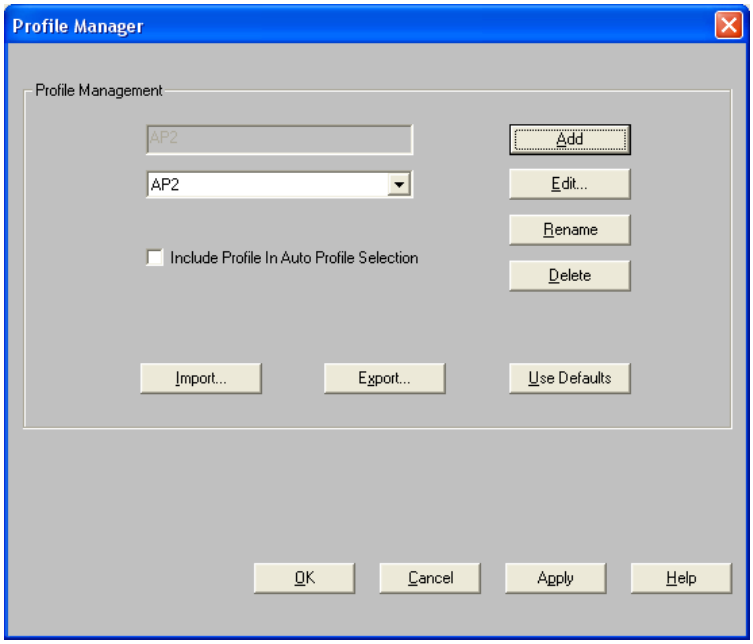


Figure 9: Selecting a profile

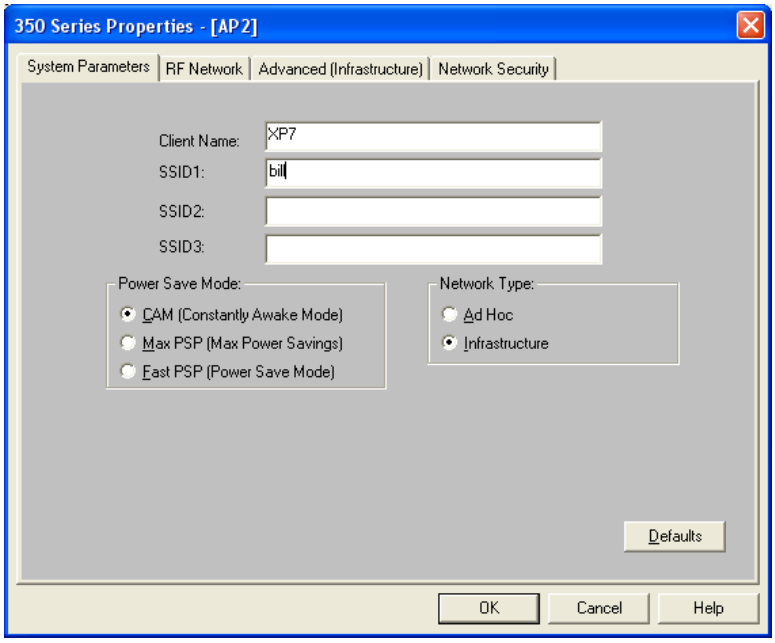


Figure 10: Editing the SSID

The IP address of the wireless card on the host can be setup by right-clicking on the Wireless Network Connection within Network Connections (Figure 11). After which the IP properties can be defined (Figure 12 – which sets up the IP address of [192.168.2.2]). If a Cisco 350 wireless card is used, the connection properties can then be displayed (Figure 13), after which the client will associated with the access point (Figure 14).

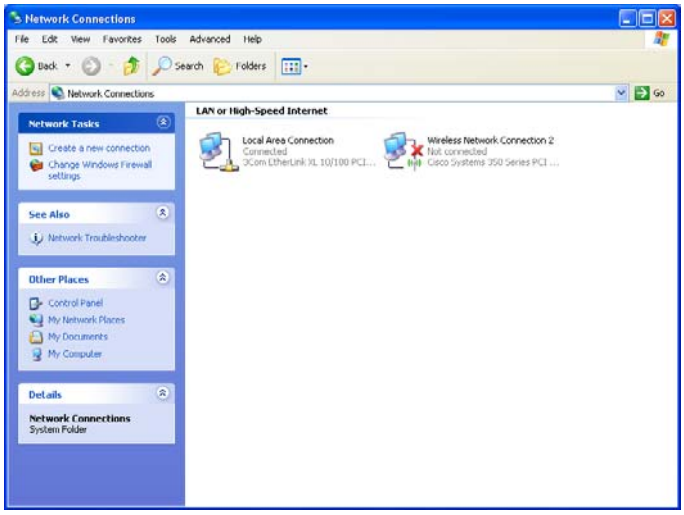


Figure 11: Setting up the wireless properties of the host

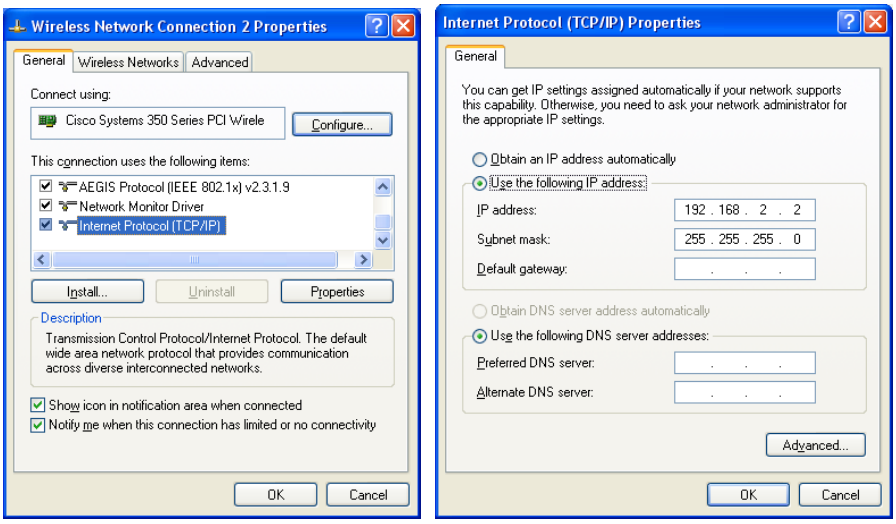


Figure 12: Setting up the wireless properties of the host

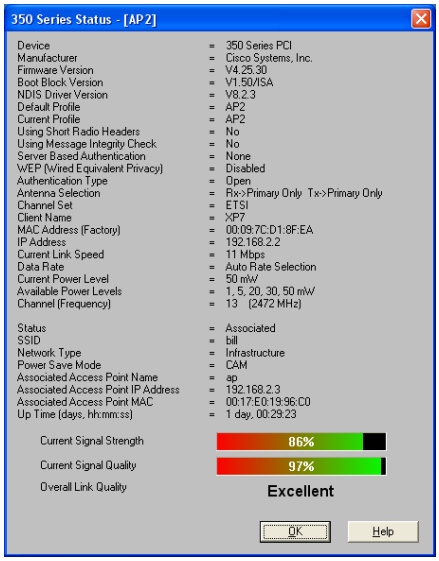


Figure 13: Cisco Aironet 350 status

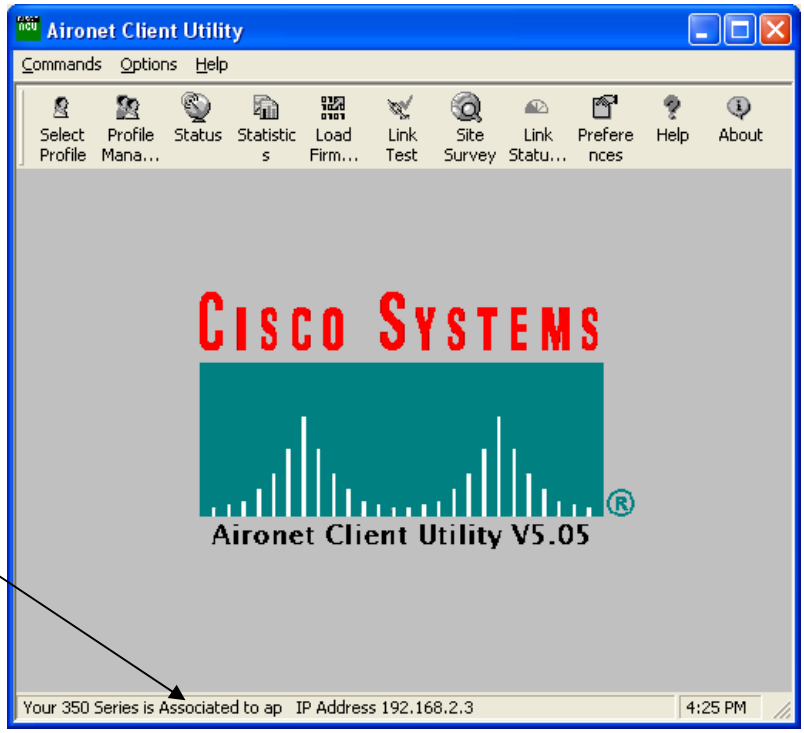


Figure 14: Cisco Aironet 350 association

If the access point is associated, the client should be able to ping the access point, such as:

```
C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=2ms TTL=150
Reply from 192.168.2.3: bytes=32 time=1ms TTL=150
Reply from 192.168.2.3: bytes=32 time=1ms TTL=150
Reply from 192.168.2.3: bytes=32 time=1ms TTL=150

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

1.2.3 Connection to the Windows 2003 server

The Windows 2003 server contains two Ethernet cards, one which connects to it the main Ethernet network, and the other connects it to the wireless network. The wireless connection can be used to setup things such as a Web server, a DHCP server, a RADIUS server, a Tacacs+ server, and so on. To make a connection from one of the hosts, use a remote desktop connection to 192.168.1.5, such as using **mstsc** (the remote desktop program), such as shown in Figure 15 and Figure 16. Figure 17 then shows the login to the server (using the login of **c072047** and the password of **co72047**). The IP address of the second Ethernet card can then be set to be part of the wireless network address range [192.168.2.5], as shown in Figure 18, after which is access point and client should be able to be pinged (Figure 19).

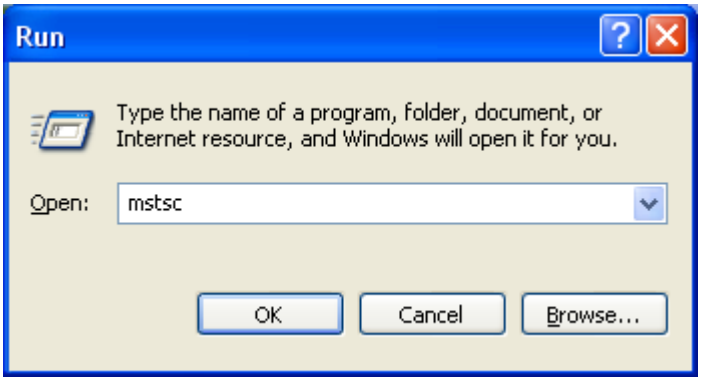


Figure 15: Running the remote desktop program

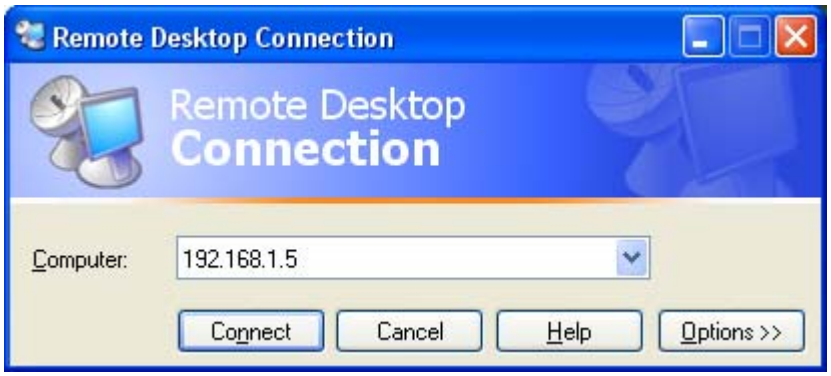


Figure 16: Connection to Windows 2003 server

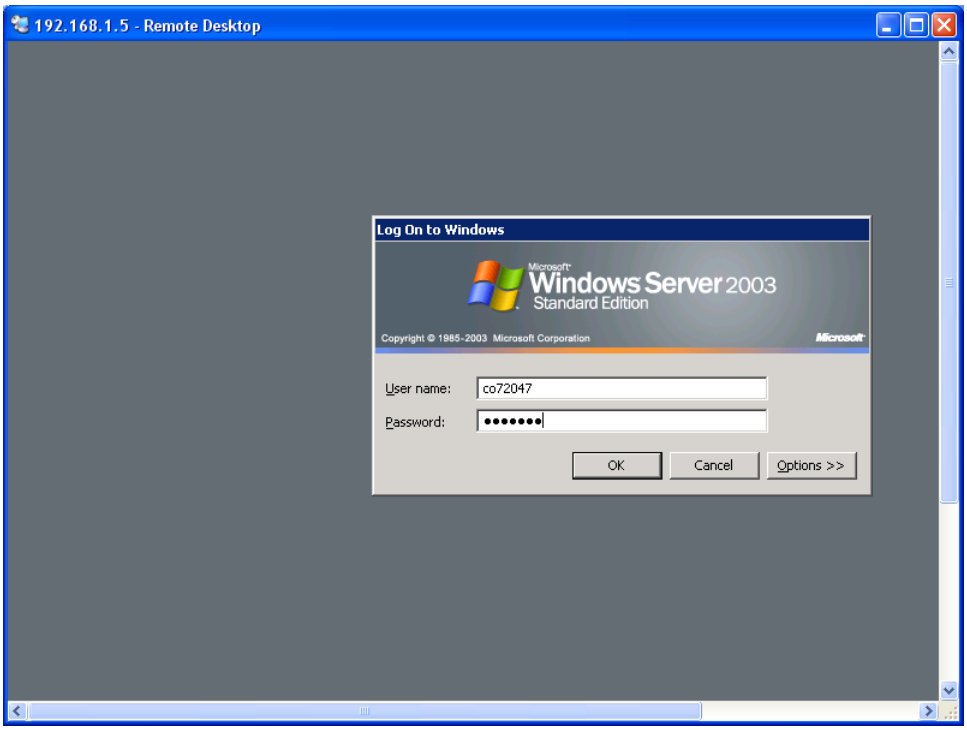


Figure 17: Login to the Windows 2003 server

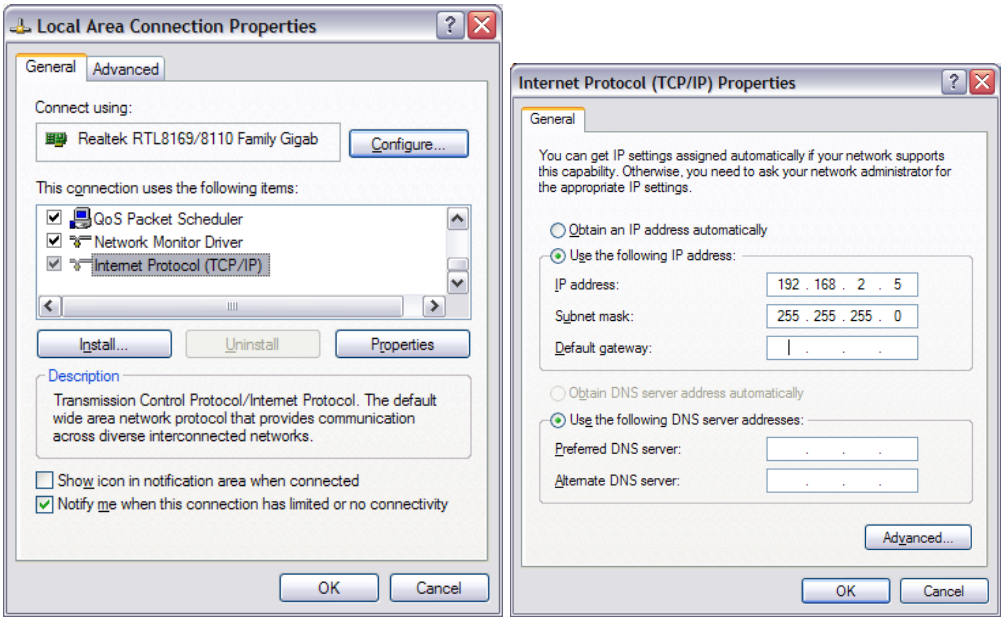


Figure 18: Setup of the IP address on the 2nd Ethernet card on the Windows 2003 server

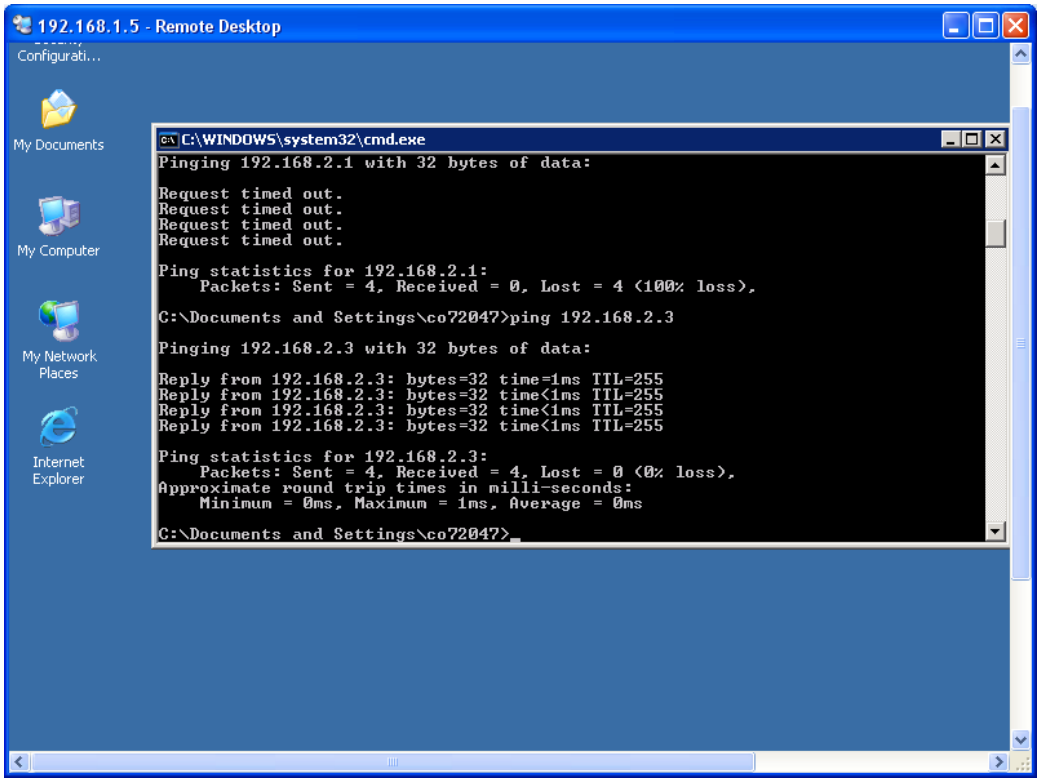


Figure 19: Completion of the login to the Windows 2003 server

1.3 Appendix 1 (Belkin card connection)

First locate the Wireless card panel, and set the Network SSID and Channel (Figure A.1). Next locate the Wireless card in Network Connections, and remove the firewall. Next right-click on the wireless icon, and set the TCP/IP settings (from Internet Protocols TCP/IP), as shown in Figure A.2. After this set the IP address of the card to one which joins onto the subnet (Figure A.3).

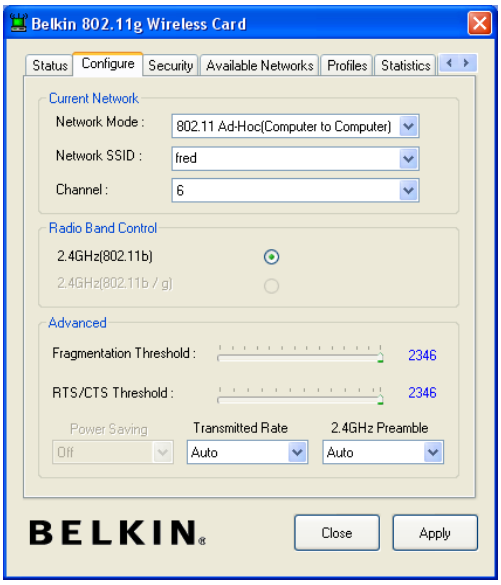


Figure A.1: Wireless card settings

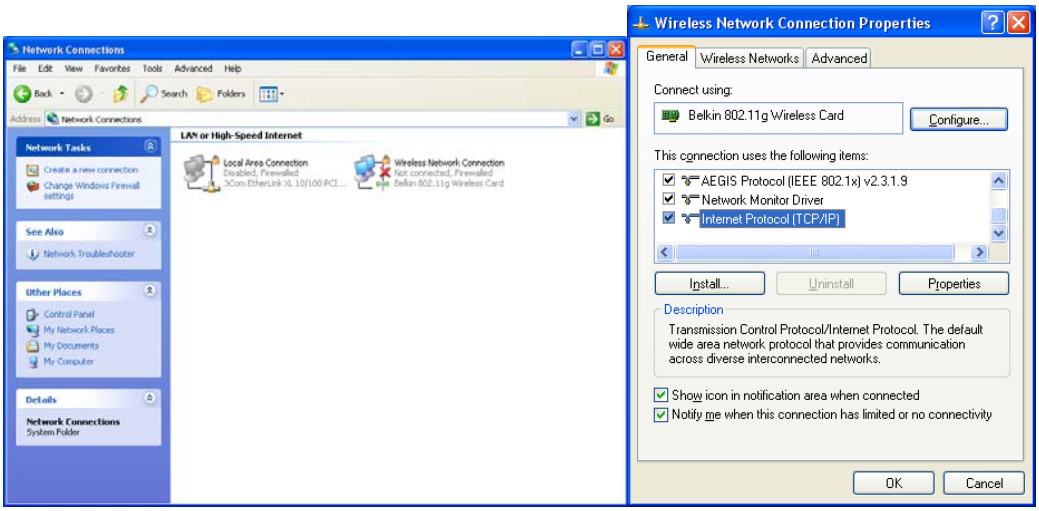


Figure A.2: Wireless card settings

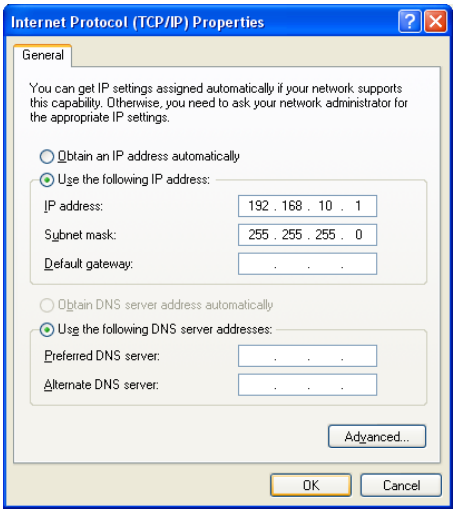
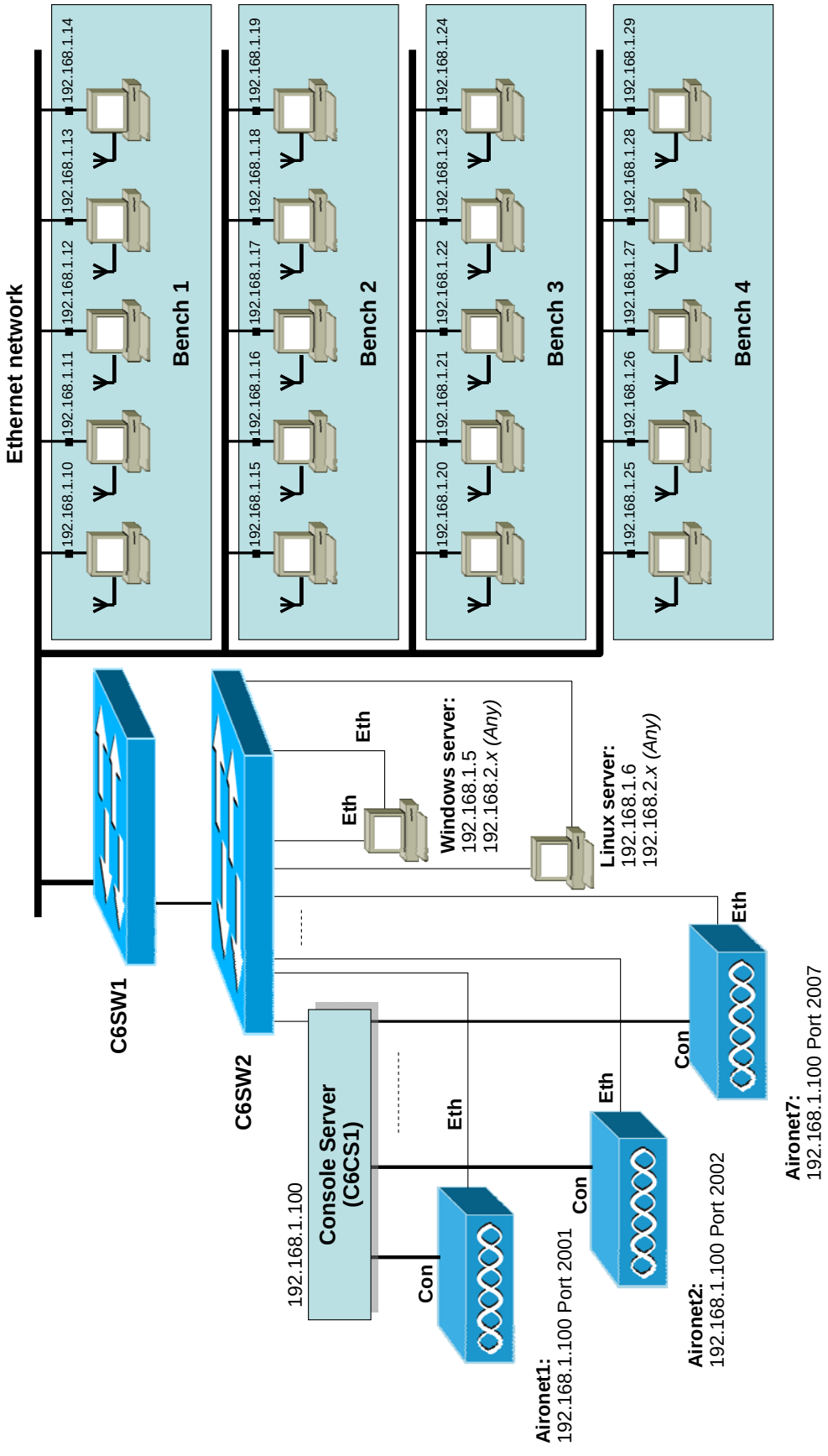


Figure A.3: Wireless IP settings

1.4 Appendix (Overall schematic)



The connections to **C6CS1** are:

Port 1: Aironet1 Console port [192.168.1.100 Port 2001]
Port 2: Aironet2 Console port [192.168.1.100 Port 2002]
Port 3: Aironet3 Console port [192.168.1.100 Port 2003]
Port 4: Aironet4 Console port [192.168.1.100 Port 2004]
Port 5: Aironet5 Console port [192.168.1.100 Port 2005]
Port 6: Aironet6 Console port [192.168.1.100 Port 2006]
Port 7: Aironet7 Console port [192.168.1.100 Port 2007]

The connections to **C6SW2** are:

FA0/1: Aironet1 FA Ethernet port
FA0/2: Aironet2 FA Ethernet port
FA0/3: Aironet3 FA Ethernet port
FA0/4: Aironet4 FA Ethernet port
FA0/5: Aironet5 FA Ethernet port
FA0/6: Aironet6 FA Ethernet port
FA0/7: Aironet7 FA Ethernet port
FA0/8: Reserved
FA0/9: Reserved
FA0/10: Windows 2003 2nd Ethernet port – used to connect to the wireless network
FA0/11: Linux 2nd Ethernet port – used to connect to the wireless network
FA0/11-20:
FA0/21: Console server (C6CS1)
FA0/22: Windows 2003 1st Ethernet port (192.168.1.5)
FA0/23: Linux 1st Ethernet port (192.168.1.6)

Login for Windows and Linux servers:

Login ID: co72047
Password: co72047

2 Labs

Lab 1: Access Point Tutorial

Using the Network-emulators, select the Wireless emulator, and perform the following:

1. You should start in the user mode:

```
>
```

2. Go into the EXEC mode using the enable command.

```
> enable
```

How does the prompt change?

3. From the EXEC mode go into the Global Configuration Mode, and use the hostname command to change the hostname to MyWireless.

```
# ?
# config t
(config) # hostname MyWireless
```

How does the prompt change?

4. Exit from the Global Configuration Mode using exit, and list the current running-config with show running-config.

```
(config) # exit
# show running-conf
```

Outline some of the settings in the running-config:

2.1.1 Using the show command

5. Complete the following command:

```
# ?
# show buffers
# show memory
# show stacks
# show hosts
# show arp
# show flash
# show history
# show version
# show interfaces
# show interface fa0
# show interface dot11radio0
```

Using the information from above what are the following:

Processor Board ID:

Processor Type:

Processor Clock Speed:

System image file:

Operating System Version:

File names shored in the Flash Memory:

Product/Model Number:

2.1.2 History commands

The main commands for history are:

```
# terminal ?
# terminal history ?
# terminal history size ?
# terminal history size 100
# show history
```

2.1.3 Clock commands

The main commands for clock are:

```
# clock ?
# clock set ?
# clock set 11:00 ?
# clock set 11:00 11 ?
# clock set 11:00 11 jun ?
# clock set 11:00 11 jun 2006
```

2.1.4 Programming the WAP ports

6. Program the two ports of the WAP with:

```
# config t
(config)# int ?
(config)# int fa0
(config-if)# ?
(config-if)# ip address ?
(config-if)# ip address 207.11.12.10 ?
```

```
(config-if)# ip address 207.11.12.10 255.255.255.0
(config-if)# no shutdown
(config-if)# exit
(config)# dot11 ssid fred
(config-ssid)# ?
(config-ssid)# guest-mode
(config-ssid)# exit
(config)# int dot11radio0
(config-if)# ?
(config-if)# ip address 192.168.0.1 255.255.255.0
(config-if)# station-role ?
(config-if)# station-role root
(config-if)# channel ?
(config-if)# channel 7
(config-if)# no shutdown
(config-if)# ssid fred
(config-if)# exit
(config)# exit
```

WAP is the root of the wireless network (other option: `repeater`)

Set radio channel to 7 (2.442GHz)

Ping the newly defined ports (`207.11.12.10 and 192.168.0.1`). Are they responding?

Next go back to the ports and shut them down:

```
# config t
(config)# int ?
(config)# int fa0
(config-if)# shutdown
(config-if)# exit
(config)# int dot11radio0
(config-if)# shutdown
(config-if)# exit
(config)# exit
```

Ping the newly defined ports (`207.11.12.10 and 192.168.0.1`) again. Are they responding?

To get rid of guest-mode:

```
# config t
(config)# dot11 ssid fred
(config-ssid)# no guest-mode
(config-ssid)# exit
(config-if)# exit
(config)# exit
```

7. Go to the EXEC mode, and view the running-config:

```
# show running-config
```

8. The WAP can access a domain server and DNS, using the ip name-server and ip domain-lookup commands:

```
# config t
```

```
(config)# ip ?
(config)# ip domain-name ?
(config)# ip domain-name mydomain.com
(config)# ip name-server ?
(config)# ip name-server 160.10.11.12
(config)# ip domain-lookup
(config)# ip default-gateway ?
(config)# ip default-gateway 10.11.12.11
```

Enable DNS lookup on the WAP

9. To get rid of any of these settings, insert a "no" in front of them, such as:

```
# config t
(config)# no ip domain-name mydomain.com
(config)# no ip name-server 160.10.11.12
(config)# no ip domain-lookup
(config)# no ip default-gateway 10.11.12.11
(config)# exit
# show running
```

10. Setting passwords for the line console and for telnet access:

```
# config t
(config)# line con 0
(config-line)# login
(config-line)# password fred
(config-line)# exit
(config)# line vty 0 15
(config-line)# login
(config-line)# password fred
(config-line)# exit
(config)# exit
```

11. Setting up a WWW server on the wireless access point:

```
# config t
(config)# ip http server
(config)# exit
# show running
```

12. If we need to change the port and the max number of connections on the WWW server:

```
# config t
(config)# ip http port 8080
(config)# ip http max-connections 2
(config)# exit
# show running
```

13. And to disable the WWW server:

```
# config t
(config)# no ip http server
(config)# exit
# show running
```

14. Setting up a user on the wireless access point:

```
# config t
(config)# username ?
(config)# username fred ?
(config)# username fred password bert
(config)# exit
# show running
```

15. To get rid of a user:

```
# config t
(config)# no username fred password bert
(config)# exit
# show running
```

16. To setup the host table on the wireless access point:

```
# config t
(config)# ip host freds 172.14.10.11
(config)# ip host berts 172.14.10.12
(config)# ip host slappi 10.15.1.100
```

17. It is possible to run a DHCP server to assign IP parameters to wireless nodes:

```
# config t
(config)# ip ?
(config)# ip dhcp ?
(config)# ip dhcp pool socpool
(config-dhcp)# ?
(config-dhcp)# network 192.168.0.0 255.255.255.0
(config-dhcp)# lease 10
(config-dhcp)# exit
(config)# exit
# show running-config
```

Sets the range of addresses to be allocated, and sets the lease for 10 days

18. Then to get rid of DHCP:

```
# config t
(config)# no ip dhcp pool socpool
(config)# exit
# show running-config
```

19. To create a banner:

```
# config t
(config)# banner motd # hello #
(config)# exit
# show running
```

20. To get rid of the banner:

```
# config t
(config)# no banner motd # hello #
```

21. To set the ARP method:

```
# config t
(config)# int dot11radio0
(config-if)# arp ?
(config-if)# arp arpa
```

22. CDP (Cisco Discovery Protocol) is set with the following:

```
# config t
(config)# cdp ?
(config)# cdp holdtime 120
(config)# cdp timer 50
(config)# end
```

Using the show cdp command, determine the settings for CDP:

23. To enable CDP on the WAP:

```
# config t
(config)# cdp run
(config)# end
```

24. To enable CDP on an interface:

```
# config t
(config)# int fa0
(config-if)# cdp enable
(config-if)# end
```

25. To show CDP information:

```
# show cdp neighbors
# show cdp neighbors detail
# show cdp neighbors traffic
```

26. To setup a local hosts table:

```
(config)# ip host LAB_A 192.5.5.1
(config)# ip host LAB_B 201.100.11.2
(config)# ip host LAB_C 223.8.151.1
(config)# ip host LAB_D 210.93.105.1
(config)# ip host LAB_E 210.93.105.2
(config)# exit
# show hosts
# show running
```

Lab 2: Access-point Tutorial

27. The power level of the access point can be set with the power command, and the speed can be set with the speed command:

```
# config t
(config)# int dot11radio0
(config-if)# power ?
(config-if)# power local ?
(config-if)# power local 30
(config-if)# power client 10
(config-if)# speed ?
(config-if)# speed 1.0
(config-if)# exit
(config)# exit
```

The access point can be used to set the power levels of the clients (in this case, 10mW)

Using the information from above what are the following:

Available power levels for access point:

Available speeds for access point:

28. With world-mode, the access point adds channel carrier set information to its beacon. This allows client devices with world mode to receive the carrier set information and adjust their settings automatically. World mode is disabled by default, to enable it:

```
# config t
(config)# int dot11radio0
(config-if)# ?
(config-if)# world-mode
(config-if)# exit
(config)# exit
```

29. The antenna can be set for both the transmit and receive options. These can be :

- **Diversity**. With this the WAP uses the antenna in which the best signal is being received.
- **Right**. This where the antenna is on the right of the WAP, and is highly directional.
- **Left**. This where the antenna is on the left of the WAP, and is highly directional.

```
# config t
(config)# int dot11radio0
(config-if)# antenna ?
(config-if)# antenna transmit ?
(config-if)# antenna transmit diversity
(config-if)# antenna receive left
(config-if)# exit
```

```
(config)# exit
```

30. The WAP can be setup to transmit a beacon signal on which devices can connect to (using a delivery traffic indication message - DTIM). The time period on which it transmits is defined in Kilomicroseconds, which is 1 millisecond (one thousands of a second). For example to set the beacon period to once every second:

```
# config t
(config)# int dot11radio0
(config-if)# beacon ?
(config-if)# beacon period ?
(config-if)# beacon period 1000
(config-if)# exit
(config)# exit
```

To get rid of the beacon signal:

```
# config t
(config)# int dot11radio0
(config-if)# no beacon period 1000
(config-if)# exit
(config)# exit
```

31. **PAYLOAD-ENCAPSULATION**. If packets are received which are not defined in IEEE 802.3 format, the WAP must format them using the required encapsulation. The methods are:

- 802.1H (**dot1h**). This is the default, and is optimized for Cisco Aironet wireless products.
- RFC1042. This is used by many wireless manufacturers (SNAP), and is thus more compatible than 802.1H.

For example:

```
# config t
(config)# int dot11radio0
(config-if)# payload-encapsulation ?
(config-if)# payload-encapsulation rfc1042
(config-if)# exit
(config)# exit
```

32. **CARRIER TEST**. The WAP can show the activity on certain channels using the carrier busy test (note that the connections to devices are dropped for about 4 seconds when these tests are made).

For example:

```
# show dot11 ?
# show dot11 carrier ?
# show dot11 carrier busy
```

33. **RTS**. The RTS (Ready To Send) is used to handshake data between the client and the WAP. RTS threshold is used to set the packet size at which the access point issues a request to send (RTS) before sending the packet. Low RTS Threshold values are useful in areas where there are many clients, or where the clients are far apart and cannot reach each other (the hidden node problem). The Maximum RTS Retries (1-128) defines the maximum number of times the access point issues an RTS before abandoning the send. For example to set the threshold at 1000 Bytes and the number of retries to 10:

```
# config t
(config)# int dot11radio0
(config-if)# rts ?
(config-if)# rts threshold ?
(config-if)# rts threshold 1000
(config-if)# rts retries ?
(config-if)# rts retries 10
(config-if)# exit
(config)# exit
```

To set the preamble to short:

```
# config t
(config)# int dot11radio0
(config-if)# preamble-short
(config-if)# exit
(config)# exit
```

To get rid of it:

```
# config t
(config)# int dot11radio0
(config-if)# no preamble-short
(config-if)# exit
(config)# exit
```

34. **PACKET RETRIES**. The maximum data retries value (1-128) defines the number of attempts that a WAP makes before dropping the packet.

```
# config t
(config)# int dot11radio0
(config-if)# packet retries 5
(config-if)# exit
(config)# exit
```

35. **FRAGMENT-THRESHOLD**. The fragmentation threshold value sets the size at which packets are fragmented (256 B to 2338 B). Low values are good when there are many errors in the transmitted data, as there will be more chance that each of the fragments will be received correctly. An example is:

```
# config t
(config)# int dot11radio0
(config-if)# fragment-threshold 1000
(config-if)# exit
(config)# exit
```

36. **IP PROXY-MOBILE**. This command is applied to the interface command to enable proxy Mobile IP operations. For example:

```
# config t
(config)# int dot11radio0
(config-if)# ip proxy-mobile
(config-if)# exit
(config)# exit
```

The basic details of the wireless access point is:

FA0 - Fast Ethernet connection to the network.
DOT11RADIO0 - 2.4GHz radio connection.
DOT11RADIO1 - 5GHz radio connection.

37. A particular problem can be were there are too many associations with the wireless device. To limit the number of associations, the max-association value is set. For example to set the maximum number of associations to 20:

```
# config t
(config)# int d0
(config-if)# ssid fred
(config-if-ssid)# ?
(config-if-ssid)# max-associations ?
(config-if-ssid)# max-associations 20
(config-if-ssid)# exit
```

38. To determine wireless nodes that have been associated with the WAP:

```
# show dot11 ?
# show dot11 associations
# show dot11 statistics client-traffic
```

What is the IP address and the MAC address of the node has been associated with the WAP:

What is the transmitted signal strength:

What is the signal quality:

39. To list controllers

```
# show controllers
```

```
!
interface Dot11Radio0
```

```
Radio 350 Series, Address 0007.50d5.bf4c, BBlock version 1.59, Software
     version 5.30.17
Serial number: vms061904jc
Carrier Set: EMEA (EU)
Current Frequency: 2452 Mhz  Channel 9
Allowed Frequencies: 2412(1) 2417(2) 2422(3) 2427(4) 2432(5) 2437(6) 2442(7)
     2447(8) 2452(9) 2457(10) 2462(11) 2467(12) 2472(13)
Current Power: 50 mW
Allowed Power Levels: 1 5 20 30 50
Current Rates:  basic-1.0 basic-2.0 basic-5.5 basic-11.0
Allowed Rates:  1.0 2.0 5.5 11.0
Best Range Rates:  basic-1.0 2.0 5.5 11.0
Best Throughput Rates:  basic-1.0 basic-2.0 basic-5.5 basic-11.0
Default Rates:  no
Radio Management (RM) Configuration:
      Beacon State      1     RM Tx Setting Enabled FALSE
      RM Tx Power Level 0     RM Tx Channel Number  0
      Saved Tx Power    0     Saved Tx Channel      0
Priority 0 cw-min 5 cw-max 10 fixed-slot 6
Priority 1 cw-min 5 cw-max 10 fixed-slot 2
Priority 2 cw-min 4 cw-max 5 fixed-slot 1
Priority 3 cw-min 3 cw-max 4 fixed-slot 1
Radio running mobile:  temp 0 C tx_power 50 bb_code 0x0
    rssi_threshold 0x0 last alarm code 0x0 gain offset 0
```

40. **SHOW CONTROLLERS**. The Show Controllers Dot11Radio0 command is used to show the status of radio interface. For example:

```
# show controllers dot11radio0
```

An example of the output is:

```
!
interface Dot11Radio0
Radio 350 Series, Address 0007.50d5.bf4c, BBlock version 1.59, Software version
5.30.17
Serial number: vms061904jc
Carrier Set: EMEA (EU)
Current Frequency: 2432 Mhz  Channel 5
Allowed Frequencies: 2412(1) 2417(2) 2422(3) 2427(4) 2432(5) 2437(6) 2442(7)
2447(8) 2452(9) 2457(10) 2462(11) 2467(12) 2472(13)
Current Power: 50 mW
Allowed Power Levels: 1 5 20 30 50
Current Rates:  basic-1.0 basic-2.0 basic-5.5 basic-11.0
Allowed Rates:  1.0 2.0 5.5 11.0
Best Range Rates:  basic-1.0 2.0 5.5 11.0
Best Throughput Rates:  basic-1.0 basic-2.0 basic-5.5 basic-11.0
Default Rates:  no
Radio Management (RM) Configuration:
      Beacon State      1     RM Tx Setting Enabled FALSE
      RM Tx Power Level 0     RM Tx Channel Number  0
      Saved Tx Power    0     Saved Tx Channel      0
```

41. **SHOW CLIENTS**. This command is used to show the details of all the associated clients, and uses:

```
# show dot11 associations all-clients
```

An example of the output is:

```
Address           : 0003.6dff.2a51     Name             :
IP Address        : 192.168.0.11       Interface        : Dot11Radio 0
Device            :    -                Software Version :

State             : Assoc              Parent           : self
SSID              : tsunami            VLAN             : 0
Hops to Infra     : 1                  Association Id   : 3
Clients Associated: 0                  Repeaters associated: 0
Key Mgmt type     : NONE               Encryption     Rate      : 11.0
Capability       :  ShortHdr
Supported Rates   : 1.0 2.0 5.5 11.0
Signal Strength   : -29  dBm           Connected for    : 913 seconds
Signal Quality    : 81 %               Activity Timeout : 31 seconds
Power-save        : Off                Last Activity    : 28 seconds ago

Packets Input     : 143                Packets Output   : 5
Bytes Input       : 16801              Bytes Output     : 266
Duplicates Rcvd   : 0                  Data Retries     : 0
Decrypt Failed    : 0                  RTS Retries      : 0
MIC Failed        : 0
MIC Missing       : 0
```

42. **SHOW DOT11 ASSOCIATIONS STATISTICS**. This command shows the statistics for the associations. For example:

```
# show dot11 associations statistics
```

An example of the output is:

```
---- DOT11 Assocation Statistics -------------

On Interface Dot11Radio0:
cDot11AssStatsAssociated        :2
cDot11AssStatsAuthenticated     :2
cDot11AssStatsRoamedIn          :0
cDot11AssStatsRoamedAway        :0
cDot11AssStatsDeauthenticated   :1
cDot11AssStatsDisassociated     :1
cur_bss_associated              :1
cur_associated                  :1
```

```
cur_bss_repeaters               :0
cur_repeaters                   :0
cur_known_ip                    :1
dot11DisassociateReason         :2
dot11DisassociateStation        :0003.6dff.2a51
dot11DeauthenticateReason       :2
dot11DeauthenticateStation      :0003.6dff.2a51
dot11AuthenticateFailStatus     :0
dot11AuthenticateFailStation    :0000.0000.0000
```

43. **SHOW INTERFACES DOT11RADIO0 STATISTICS**. This command shows the statistics for the radio port. For example:

```
# show interfaces dot11radio0 statistics
```

An example of the output is:

```
DOT11 Statistics (Cumulative Total/Last 5 Seconds):
RECEIVER                                TRANSMITTER
Host Rx Bytes:          41758 /   0     Host Tx Bytes:          135270 /   0
Unicasts Rx:              450 /   0     Unicasts Tx:              1258 /   0
Unicasts to host:         450 /   0     Unicasts by host:           11 /   0
Broadcasts Rx:           1247 /   0     Broadcasts Tx:           30329 /  49
Beacons Rx:                 0 /   0     Beacons Tx:              29773 /  49
Broadcasts to host:         0 /   0     Broadcasts by host:        556 /   0
Multicasts Rx:              0 /   0     Multicasts Tx:              77 /   0
Multicasts to host:         0 /   0     Multicasts by host:         77 /   0
Mgmt Packets Rx:         1247 /   0     Mgmt Packets Tx:         1247 /   0
RTS received:               0 /   0     RTS transmitted:            0 /   0
Duplicate frames:          65 /   0     CTS not received:           0 /   0
CRC errors:                57 /   0     Unicast Fragments Tx:    1258 /   0
WEP errors:                 0 /   0     Retries:                    0 /   0
Buffer full:                0 /   0     Packets one retry:          0 /   0
Host buffer full:           0 /   0     Packets > 1 retry:          0 /   0
Header CRC errors:        656 /   0     Protocol defers:            0 /   0
Invalid header:             0 /   0     Energy detect defers:      52 /   0
Length invalid:             0 /   0     Jammer detected:            0 /   0
Incomplete fragments:       0 /   0     Packets aged:               0 /   0
Rx Concats:                 0 /   0     Tx Concats:                 0 /   0

RATE 11.0 Mbps
Rx Packets:               450 /   0     Tx Packets:                 8 /   0
Rx Bytes:               41664 /   0     Tx Bytes:                 764 /   0
RTS Retries:                0 /   0     Data Retries:               0 /   0
```

The full list of key interfaces are:

```
# show interface ?
# show interface fa0
# show interface dot11radio0
# show interface bvi
```

44. **SHOW DOT11 NETWORK-MAP**. This command shows the radio network map. For example:

```
# show dot11 ?
# show dot11 network-map
# config t
(config)# dot11 network-map
(config)# exit
# show dot11 network-map
# show dot11 carrier ?
# show dot11 carrier busy
```

Which frequency is the most utilized:

45. A few other show commands are:

```
# show ip
# show ip ?
# show led
# show led ?
# show led flash
# show line
# show log
```

```
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0
flushes, 0 overruns)
    Console logging: level debugging, 31 messages logged
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 32 messages logged
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
       Trap logging: level informational, 35 message lines logged


Log Buffer (4096 bytes):

*Mar  1 00:00:04.103: soap_pci_subsys_init: slot 3 found radio
*Mar  1 00:00:04.405: %LINK-5-CHANGED: Interface Dot11Radio0, changed state
to reset
*Mar  1 00:00:05.429: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Dot11Radio0, changed state to down
*Mar  1 00:00:06.432: %LINK-3-UPDOWN: Interface FastEthernet0, changed state
to up
*Mar  1 00:00:07.432: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0, changed state to up
```

```
*Mar  1 00:00:15.384: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0, changed state to down
*Mar  1 00:00:25.435: %SYS-5-RESTART: System restarted --
```

```
# show vlans
```

46. Some other show commands are:

```
# show aliases
# show caller
# show cca
# show class-map
# show clock
# show crash
# show dhcp ?
# show dot11 ?
```

```
  adjacent-ap        Display adjacent AP list
  antenna-alignment  Display recent antenna alignment results
  arp-cache          Arp Cache
  associations       association information
  carrier            Display recent carrier test results
  linktest           Display recent linktest results
  network-map        Network Map
     statistics         statistics information
```

```
# show dot11 adjacent-ap
# show dot11 arp-cache
# show dot11 associations
```

```
802.11 Client Stations on Dot11Radio0:

SSID [tsunami] :

MAC Address    IP address      Device        Name            Parent
State
0090.4b54.d83a 192.168.2.2     4500-radio    -               self
Assoc


Others:  (not related to any ssid)
```

```
# show dot11 carrier ?
# show dot11 carrier busy
# show dot11 network-map
# show dot11 statistics
# show dot11 statistics ?
# show dot11 statistics client-traffic
```

```
Clients:
3-0090.4b54.d83a pak in 372 bytes in 31151 pak out 3 bytes out 262
```

```
dup 0 decrpyt err 0 mic mismatch 0 mic miss 0
tx retries 0 data retries 0 rts retries 0
signal strength 43 signal quality 83
```

47. For radio tests:

```
# dot11 ?
# dot11 dot11radio0 ?
# dot11 dot11radio0 carrier ?
# dot11 dot11radio0 carrier busy
# dot11 dot11radio0 linktest
```

Lab 3: Ad-hoc networks

Outline:

The objective of this lab is demonstrate the principles of ad-hoc networks, especially in joining a network and in assessing its performance. At the start of the lab you will be given a name for your SSID, and assigned into groups.

What is the SSID that you group has been assigned: [GroupA] [GroupB] [GroupC] [GroupD] [GroupE] [GroupF] [GroupG]

2.1.5 Setting SSID and mode

First locate the Wireless card panel, and set the **Network Mode** to 802.11 Ad-hoc and the **Network SSID** to the name you have been assigned. The channel should also be set to Channel 6, as shown in Figure 1.

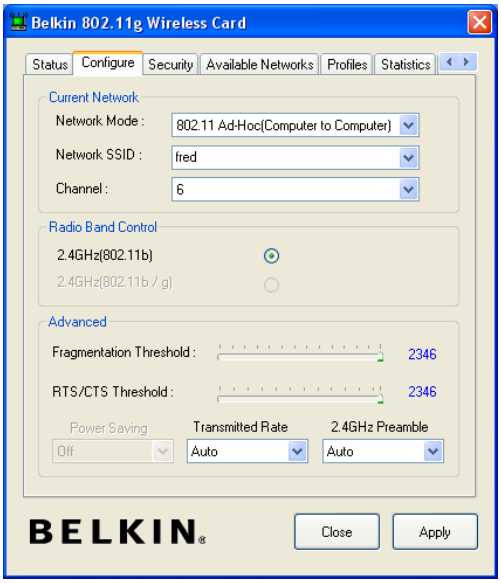


Figure 1: Wireless card settings

2.1.6 Setting IP address

Next locate the Wireless card in Network Connections, and remove the firewall. Next right-click on the wireless icon, and set the TCP/IP settings (from Internet Protocols TCP/IP), as shown in Figure 2. After this set the IP address of the card to one which joins onto the subnet (Figure 3):

192.168.10.0

Which IP address and subnet mask have you chosen, and what are the other nodes in your group assigned as: **Why does the gateway not have to be set:**

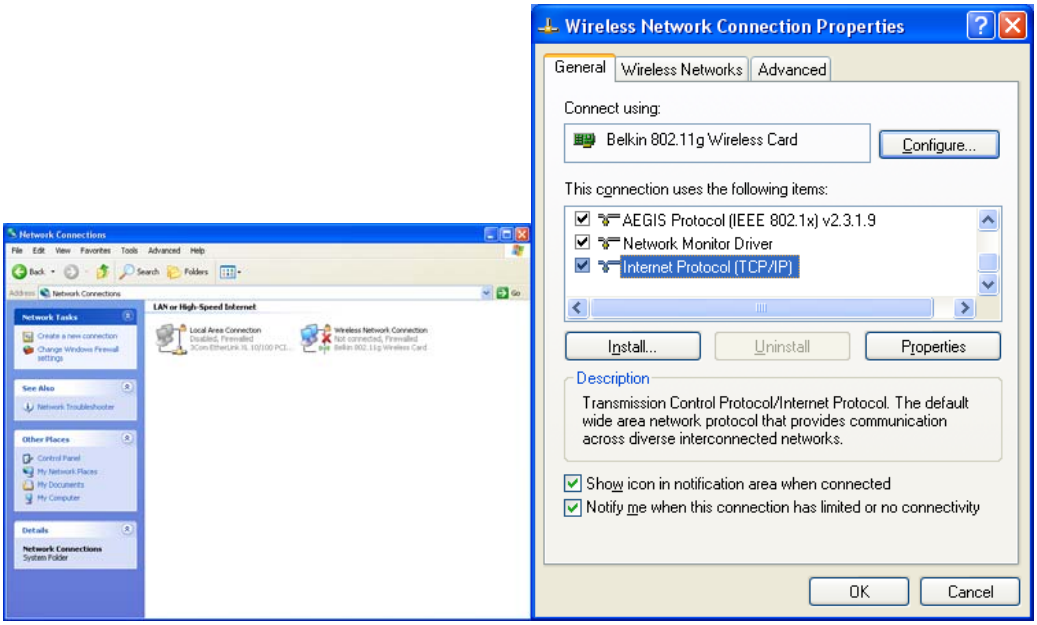


Figure 2: Wireless card settings

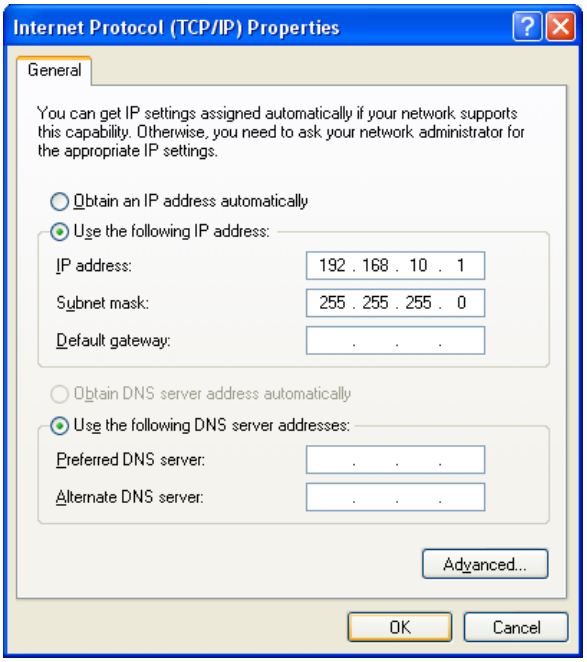


Figure 3: Wireless IP settings

2.1.7 Connect to your ad-hoc network

Next scan for the available networks, and connect to your SSID.

Was the connection successful?

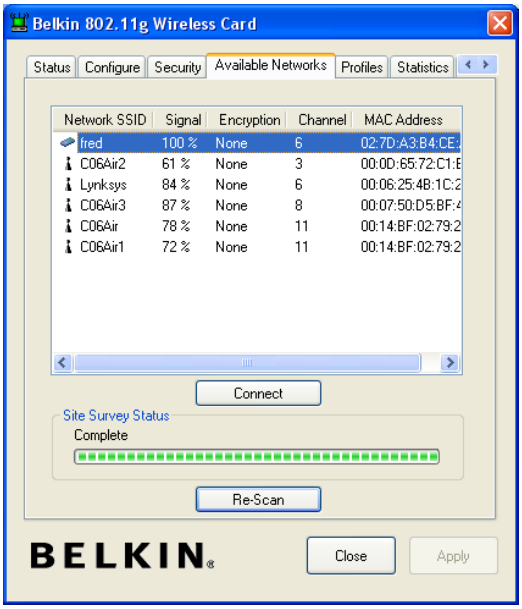


Figure 4: Scanning for ad-hoc networks

Next ping your node and the others in your network, such as with:

```
C:\Documents and Settings\co72047.XP3>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:

Reply from 192.168.10.2: bytes=32 time=1467ms TTL=128
Reply from 192.168.10.2: bytes=32 time=1ms TTL=128
Reply from 192.168.10.2: bytes=32 time<1ms TTL=128
Reply from 192.168.10.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1467ms, Average = 367ms
```

Was the ping successful to the nodes in your group:

2.1.8 Network characteristics

Next use IPCONFIG/ALL to determine the network settings of your wireless card, such as:

```
C:\>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : XP3
        Primary Dns Suffix  . . . . . . . : c06server
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Wireless Network Connection:
```

```
        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Belkin 802.11g Wireless Card
        Physical Address. . . . . . . . . : 00-11-50-15-B5-A2
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.10.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
```

What is the MAC address of your card:

What is the host name of your computer:

What is the IP address of the card:

2.1.9 Sharing a folder

On each of the machines within your network, create a folder, and share it to everyone in the network, such as shown in Figure 5, 6 and 7. Next, in Figure 8, access the shared folders within your group with the form of **\\remoteIP**. Create a document, and get someone to access it remotely.

Has the sharing been successful?

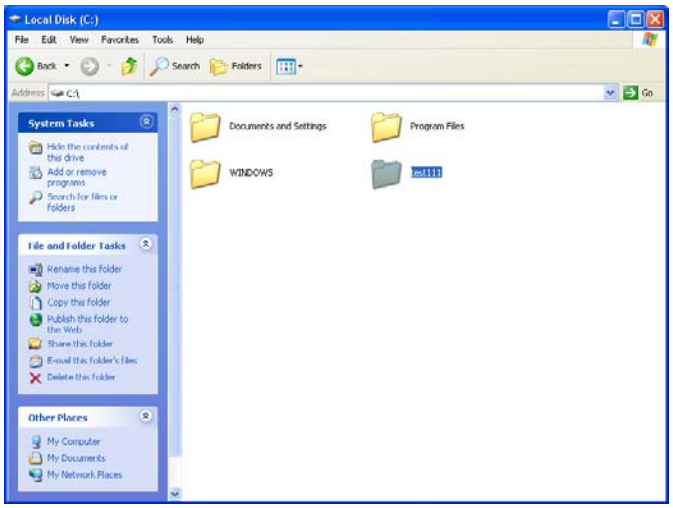


Figure 5: Creating a folder

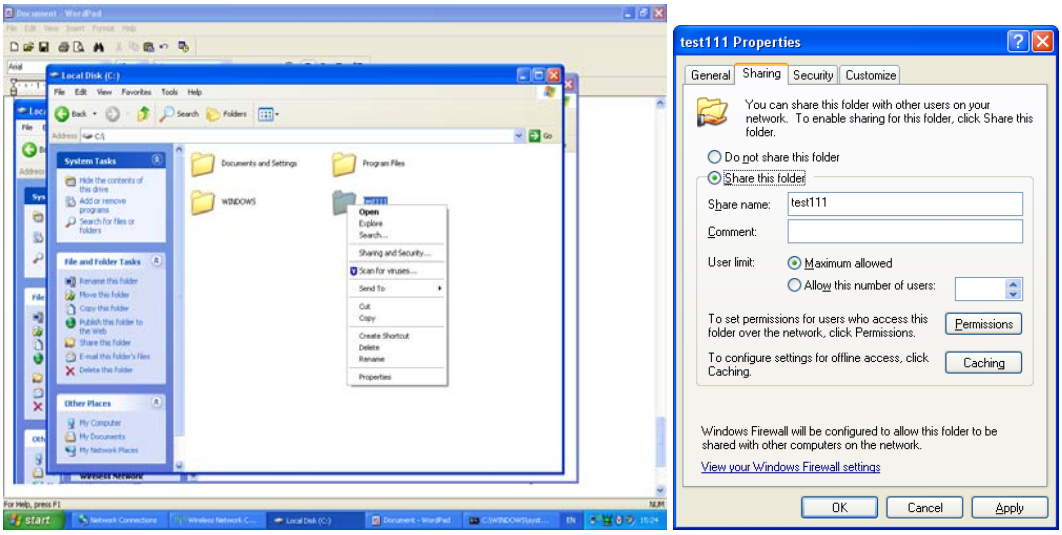


Figure 6: Sharing a folder

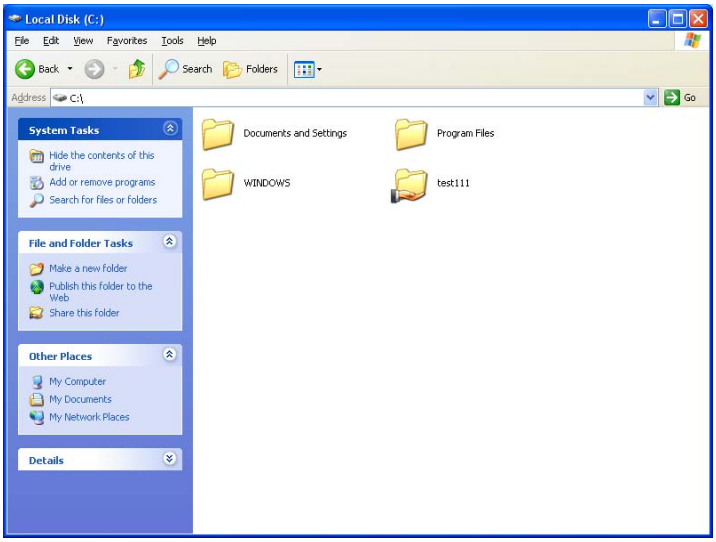


Figure 7: Sharing a folder

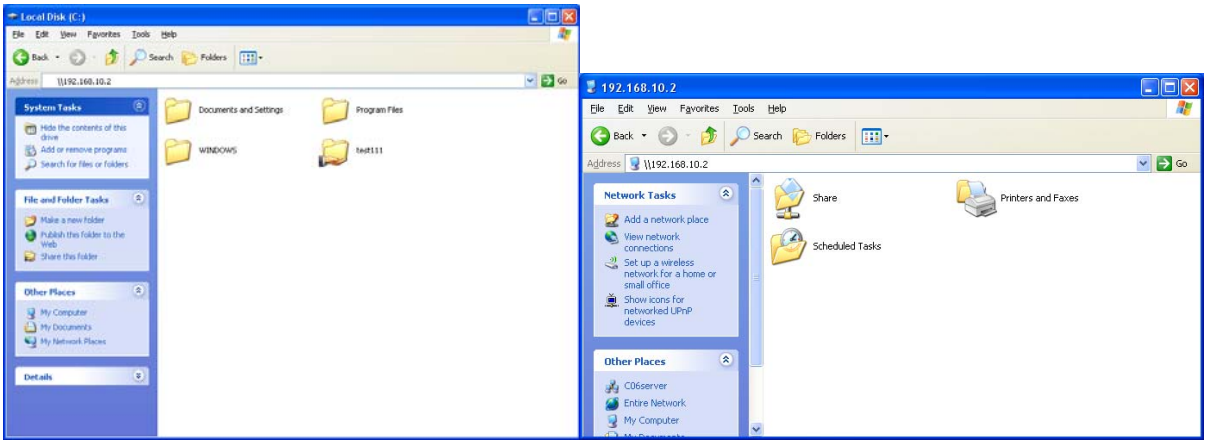


Figure 8: Accessing a shared folder

2.1.10 Viewing network traffic

Run Ethernet, as shown in Figure 9, and unset the **Capture packets in promiscuous mode**, and re-ping the network, and view the result in Ethernet (Figure 10).

Did it capture the ping event: **Outline some of the information that is provided on the ping with Ethereal:**

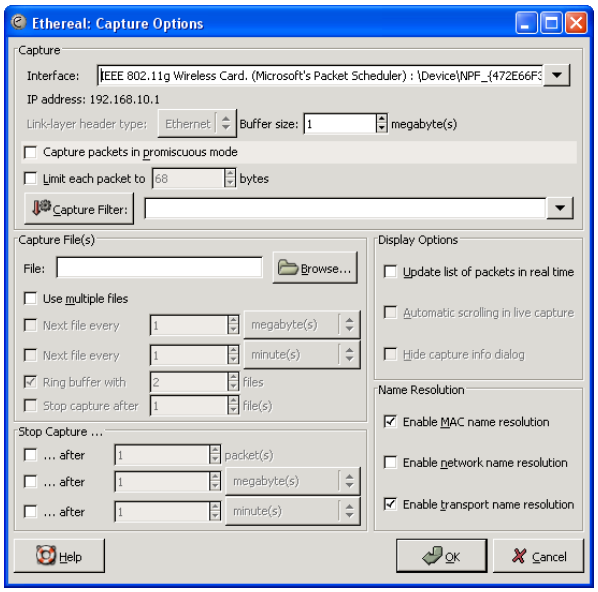


Figure 9: Ethereal

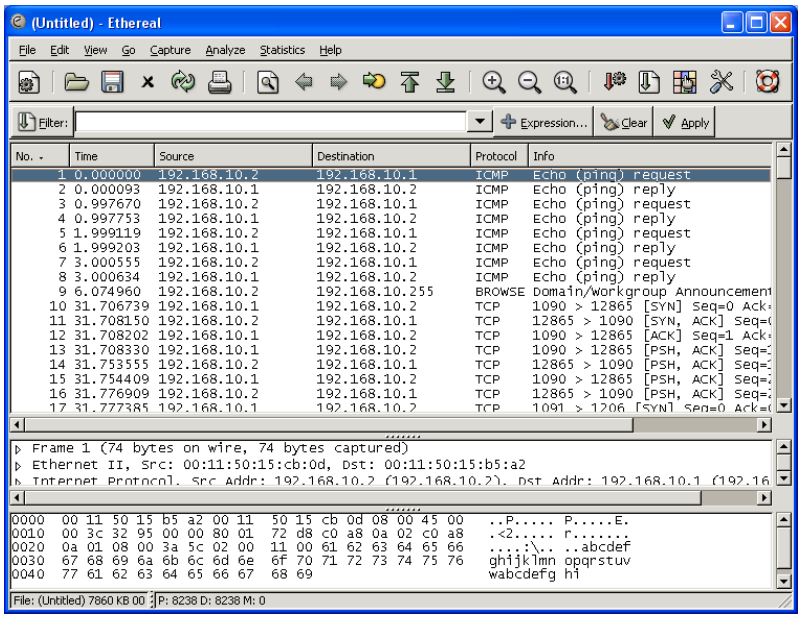


Figure 10: Ethereal

2.1.11 Running a performance test

On one machine run the **netserver** program, and, on another, run the **netperf** program, and measure the data throughput. In the following, the throughput is measured at 5.84Mbps:

```
C:\>netserver
Starting netserver at port 12865

C:\test111>netperf -H 192.168.10.2
TCP STREAM TEST to 192.168.10.2
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

  8192   8192   8192    10.00       5.84
```

What is the throughput in your test:

2.1.12 Client/server operation

Once the connection is working, the next thing to test is the TCP/Application layer. For this, run the **Basic Server** (Figure 11) on one machine, and the **Basic Client** (Figure 12) on another, and create a connection. Once connected run Ethereal and see if you can see the data transfer.

Did the devices connect:

Can you see the data packets in Ethereal and read the contents of the conversation:

Which TCP port does the server use to communicate:

Which TCP port does the client use to communicate:

To determine the open TCP ports, run the **netstat -a** command, such as:

```
C:\> netstat -a
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    XP3:epmap              0.0.0.0:0              LISTENING
  TCP    XP3:microsoft-ds       0.0.0.0:0              LISTENING
  TCP    XP3:1026               0.0.0.0:0              LISTENING
  TCP    XP3:netbios-ssn        0.0.0.0:0              LISTENING
  TCP    XP3:1068               192.168.10.2:1001      ESTABLISHED
```

Which line in your run relates to the client-server connection:

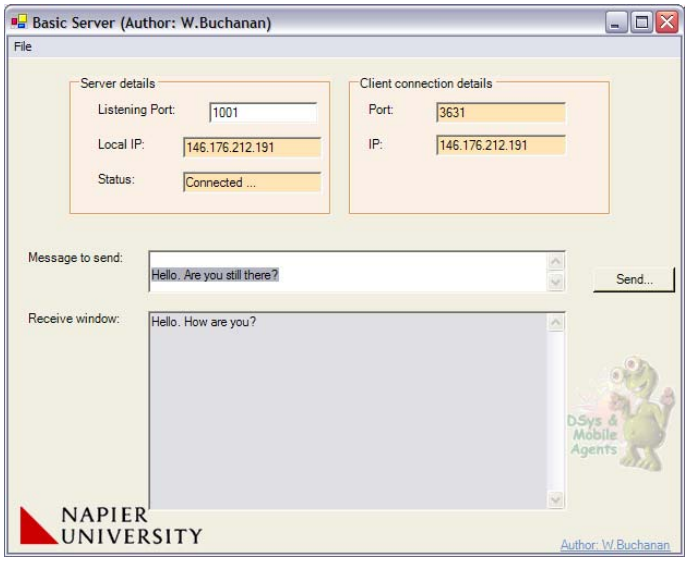


Figure 11: **Basic server**

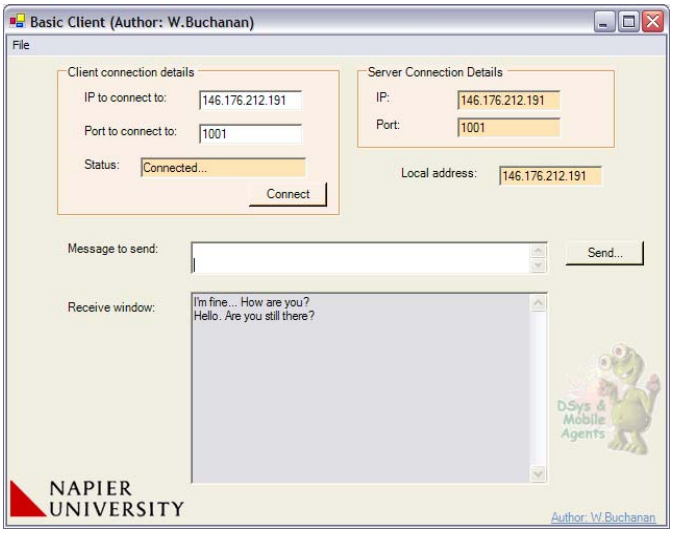


Figure 12: **Basic client**

2.1.13 Encryption (64-bit)

The next setting is to define 64-bit encryption with a pass phrase of **cisco** (left-hand side of Figure 13). Reconnect again (right-hand side of Figure 13), and verify that the network stills works. Next run the performance test again, such as:

```
C:\>netperf -H 192.168.10.2
TCP STREAM TEST to 192.168.10.2
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec
  8192   8192   8192    10.00       5.71
```

What is the throughput:

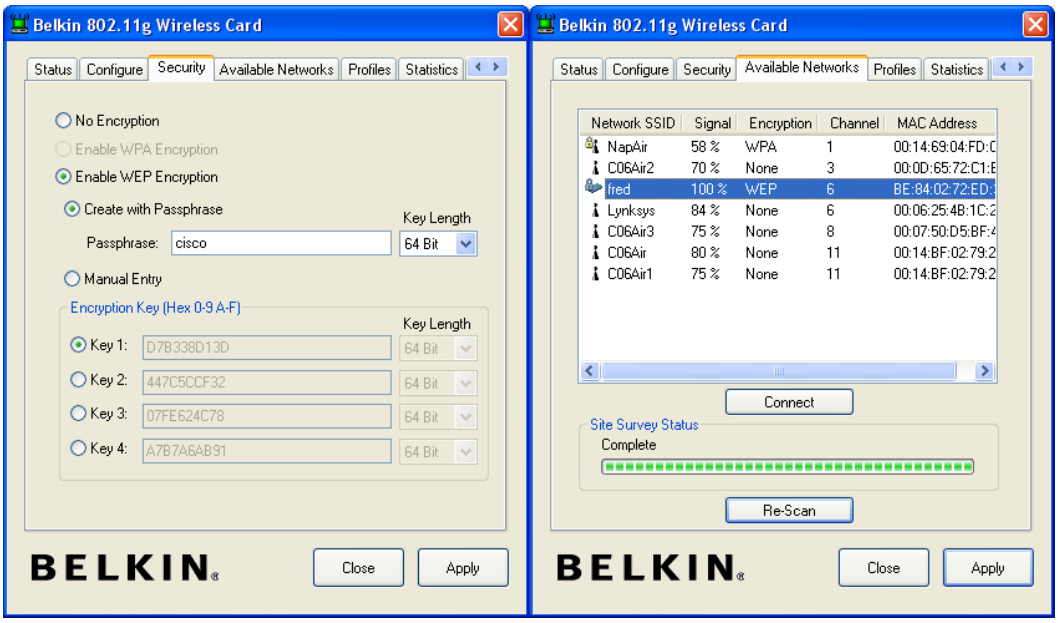


Figure 13: **64-bit WEP encryption**

2.1.14 Encryption (64-bit)

The next setting is to define 128-bit encryption with a pass phrase of **cisco** (left-hand side of Figure 14). Reconnect again (right-hand side of Figure 14), and verify that the network stills works, and measure the performance, such as:

```
C:\>netperf -H 192.168.10.2
TCP STREAM TEST to 192.168.10.2
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

  8192   8192   8192    10.00       5.85
```

What is the throughput:

From the previous results, what affect does WEP encryption have on the throughput:

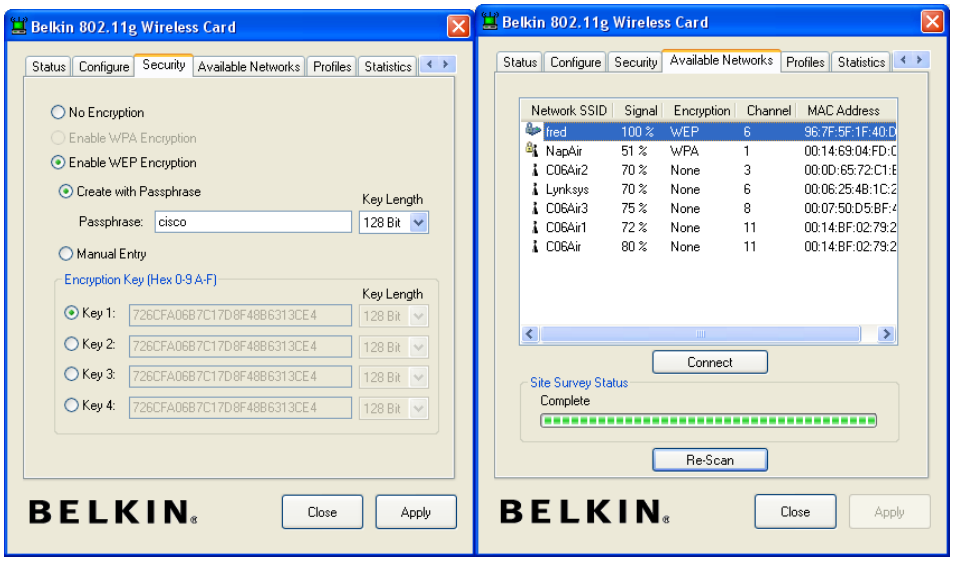


Figure 14: 128-bit WEP encryption

2.1.15 Differing connection speeds

Next we will determine if the connection speed has an affect on the throughput. For this set the transmission rate at **1Mbps** (see Figure 15), and run the performance test, such as:

```
C:\test111>netperf -H 192.168.10.2
TCP STREAM TEST to 192.168.10.2
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

  8192   8192   8192    10.00       0.81
```

What is the throughput:

What can you conclude from this:

2.1.16 Client/server and Encryption

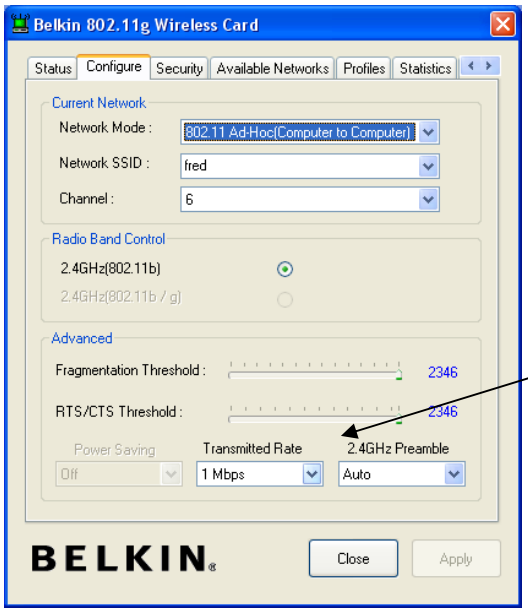
Run the client and server on different machines again, and connect. Run Ethernet, and view the network traffic.

Is it now possible to view the text which is passed between the client and server:

2.1.17 Client/server and Encryption

As a final test, change the pass phase on one of the computers.

Can this computer now communicate with the other nodes:



Transmission rate

Lab 4: Infrastructure Network

You will be assigned a group. In this lab the setup is as follows:

Group	Device	SSID	BVI	Host range	Radio channel
A	**Aironet1**	**GroupA**	**192.168.2.1**	**192.168.2.10-192.168.2.12**	**2**
B	**Aironet2**	**GroupB**	**192.168.2.2**	**192.168.2.13-192.168.2.14**	**3**
C	**Aironet3**	**GroupC**	**192.168.2.3**	**192.168.2.15-192.168.2.17**	**4**
D	**Aironet4**	**GroupD**	**192.168.2.4**	**192.168.2.18-192.168.2.19**	**5**
E	**Aironet5**	**GroupE**	**192.168.2.5**	**192.168.2.20-192.168.2.22**	**7**
F	**Aironet6**	**GroupF**	**192.168.2.6**	**192.168.2.23-192.168.2.24**	**8**
G	**Aironet7**	**GroupG**	**192.168.2.7**	**192.168.2.25-192.168.2.27**	**9**

The setup for the Windows server is 192.168.2.8 and the Linux server is 192.168.2.9. A diagram of the system is shown in Figure 1.

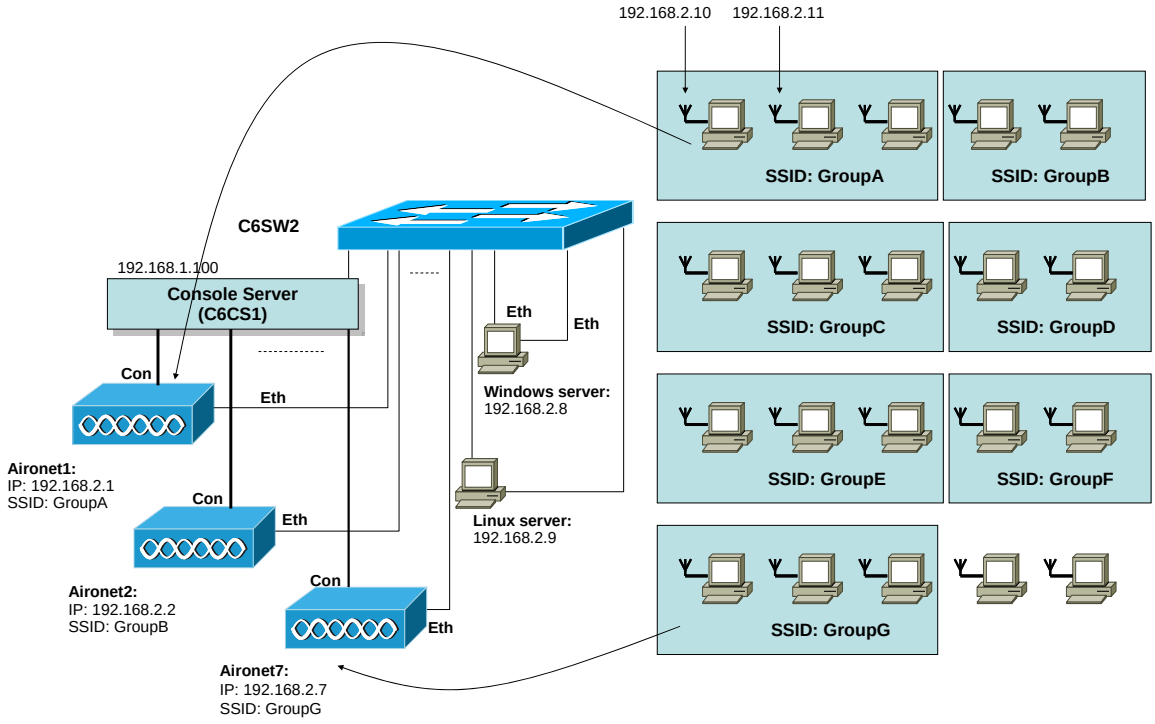


Figure 1:

An example setup for GroupA is:

```
hostname GroupA
dot11 ssid GroupA
  authentication open
  guest-mode
int bvi1
  ip address 192.168.1.1 255.255.255.0
interface d0
```

```
  channel 2
  station-role root
  ssid GroupA
  no shutdown
interface fa0
  no shutdown
```

1. Configure your Aironet for the required settings for your group.

Outline the main configuration settings:

2. Set the IP address of your wireless cards. All the hosts on your network will connect to the same subnet (192.168.2.x), as illustrated in Figure 1. What is your IP address?

What is your IP address, and what are the others in the group:

3. Scan for your SSID, and connect to it.

Can you ping your own machine: **Can you ping the access point:**

NOTE: Sometimes the card must be disabled and then enabled for it to fully re-connect.

An example of a ping is:

```
C:\Documents and Settings\co72047.XP2>ping 192.168.2.20

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time<1ms TTL=128
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128

C:\Documents and Settings\co72047.XP2>ping 192.168.1.240
Pinging 192.168.1.240 with 32 bytes of data:
Reply from 192.168.1.240: bytes=32 time=1ms TTL=255
Reply from 192.168.1.240: bytes=32 time=1ms TTL=255
```

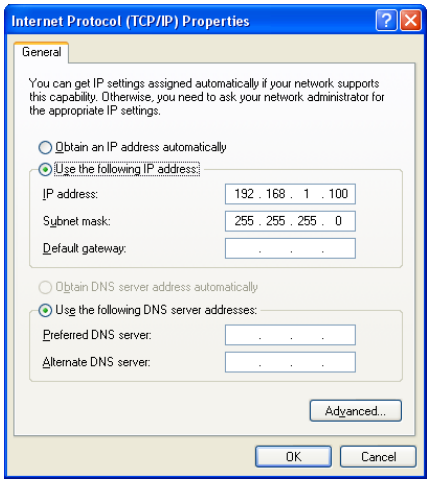


Figure 1: Wireless card settings

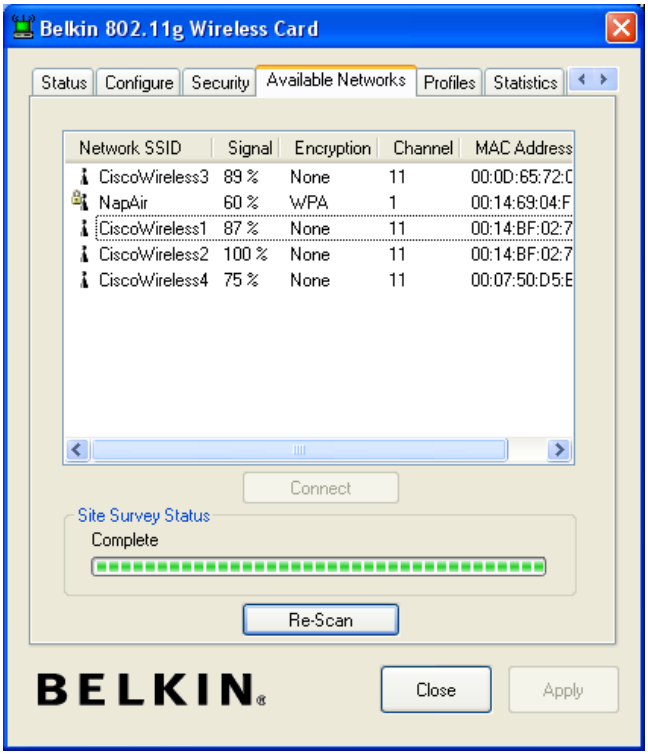


Figure 2: **SSID scan**

4. Next access the Web page of the access point with http://192.168.2.*x*:

Was the connection successful?

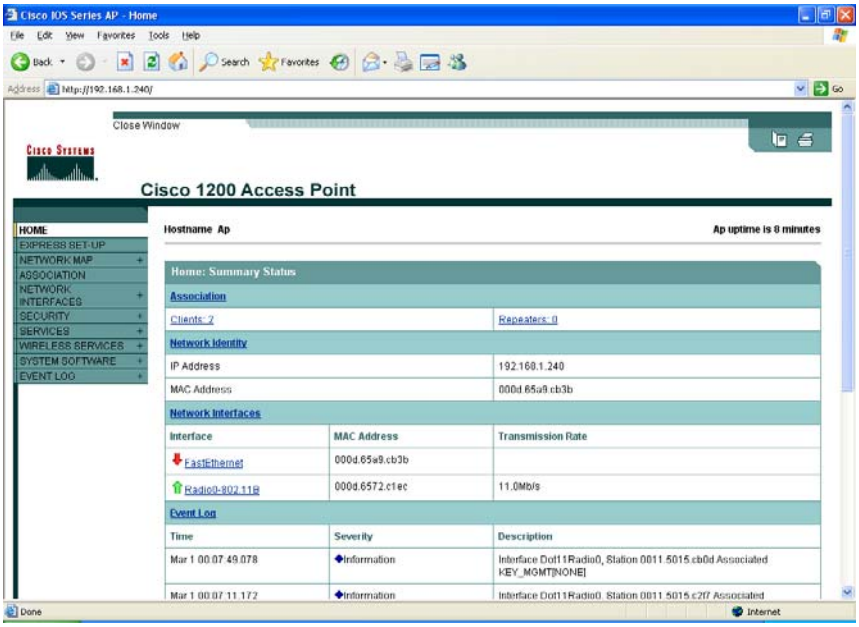


Figure 3: Aironet device home page

5. Ping all the devices in your network. Next, as with Lab 3, share a folder on your machine with the rest of your network:

Was the sharing successful:

6. As with Lab 3, run netperf and netserver, and determine the throughput of the connection between two hosts:

Network throughput:

7. As with Lab 3, run the client and server, and make a connection between two hosts:

Network throughput:

8. Once all the group have setup their wireless networks, ping all the nodes in the network:

Can you ping all the hosts? **Can you ping the Windows server (192.168.2.8)?**

9. Using **mstsc**, make a remote desktop connection to the Windows server.

Can you remote desktop to the Windows server?

Lab 5: Remote Connections

You will be assigned a group. In this lab the setup is as follows:

Group	Device	SSID	BVI	Host range	Radio channel
A	**Aironet1**	**GroupA**	**172.16.1.1**	**172.16.1.10-172.16.1.12**	**2**
B	**Aironet2**	**GroupB**	**172.16.1.2**	**172.16.1.13-172.16.1.14**	**3**
C	**Aironet3**	**GroupC**	**172.16.1.3**	**172.16.1.15-172.16.1.17**	**4**
D	**Aironet4**	**GroupD**	**172.16.1.4**	**172.16.1.18-172.16.1.19**	**5**
E	**Aironet5**	**GroupE**	**172.16.1.5**	**172.16.1.20-172.16.1.22**	**7**
F	**Aironet6**	**GroupF**	**172.16.1.6**	**172.16.1.23-172.16.1.24**	**8**
G	**Aironet7**	**GroupG**	**172.16.1.7**	**172.16.1.25-172.16.1.27**	**9**

The setup for the Windows server is 172.16.1.8 and the Linux server is 172.16.1.9. An example setup for GroupA is:

```
hostname GroupA
dot11 ssid GroupA
  authentication open
  guest-mode
int bvi1
  ip address 172.16.1.1 255.255.255.0
interface d0
  channel 2
  station-role root
  ssid GroupA
  no shutdown
interface fa0
  no shutdown
```

1. Setup your wireless network, and ping all the nodes in your network.

Can all the nodes connect to the wireless network, and can ping each other:

Use the command `show dot11 assoc` **on the access point. What is the output:**

2. Telnet is one of the most widely use protocols for remote access of devices, and uses port 23 by default. Enable up to 16 TELNET sessions on the access point with the configuration:

```
# config t
(config)# line vty 0 15
(config-line)# transport input telnet
```

3. Using the TELNET program in Windows, test if the wireless nodes can access

the wireless access point:

Can all the nodes TELNET into the access point:

4. Next, using the **PuTTY** client, TELNET into the wireless access point (as illustrated in Figure 3).

Can all the nodes TELNET into the access point:

5. Next, run Ethereal, and capture the wireless traffic, and re-TELNET into the access point. Verify that you can read the username and the password from the network traffic (Figure 4).

Can you view the username and password:

6. Next, create a number of usernames and passwords using a form such as:

```
(config)# username fred password bert
(config)# username freddy password berty
```

Using a username and password for each person in the group, login using their username and password. At the same time, using Ethereal, verify that the username and password can be determined:

Can the username and password be determined for each session:

An example login is shown next:

```
User Access Verification
Username: fred
Password: bert
ap>
```

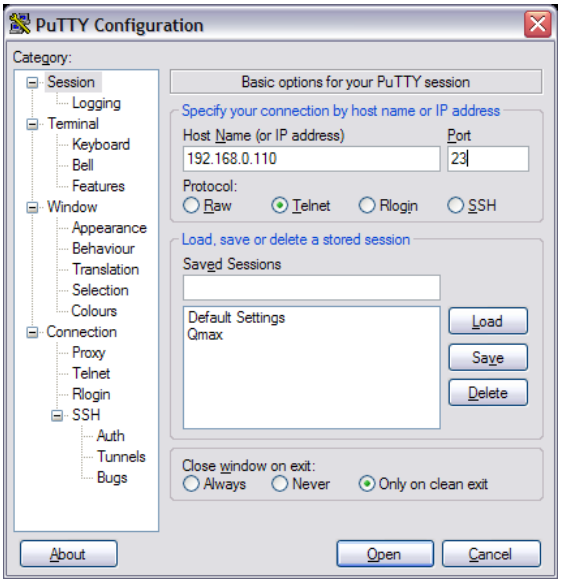


Figure 1: PuTTY connection for TELNET

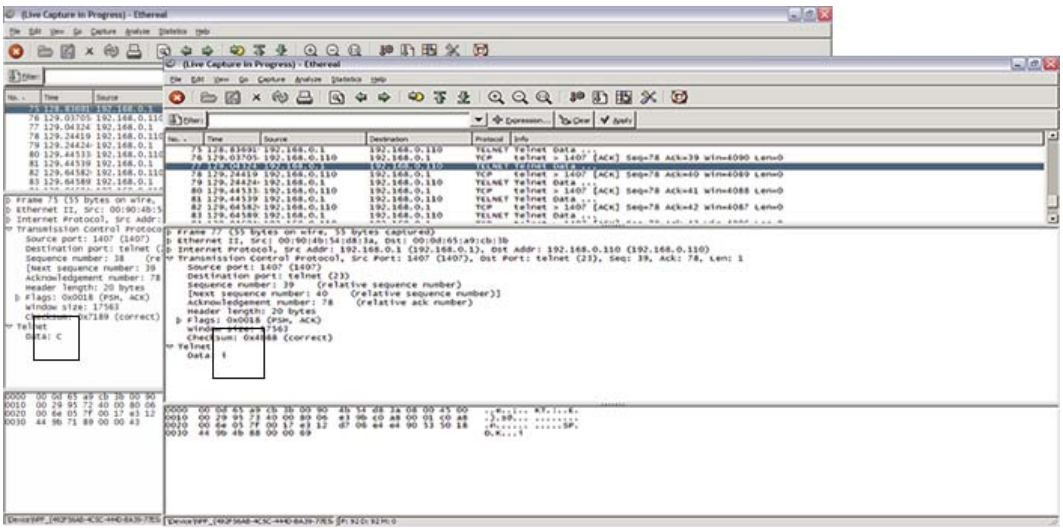


Figure 2: Ethereal showing the plaintext password

7. Telnet is seen as being an insecure, and, if security is an issue, it should be disabled as a service and replaced with SSH, which uses encryption. To enable only SSH on the access point implement the following:

```
# config t
(config)# ip domain-name fred.com
(config)# crypto key generate rsa
(config)# exit
# show ip ssh
# config t
(config)# ip ssh rsa keypair-name ap.fred.com
(config)# line vty 0 15
(config-line)# transport input ssh
```

Do the connections work:

What TCP port is used, by default: **Is TELNET access possible:**

8. Next connect to the access point using SSH (with the PuTTY client), as shown in Figure 5.

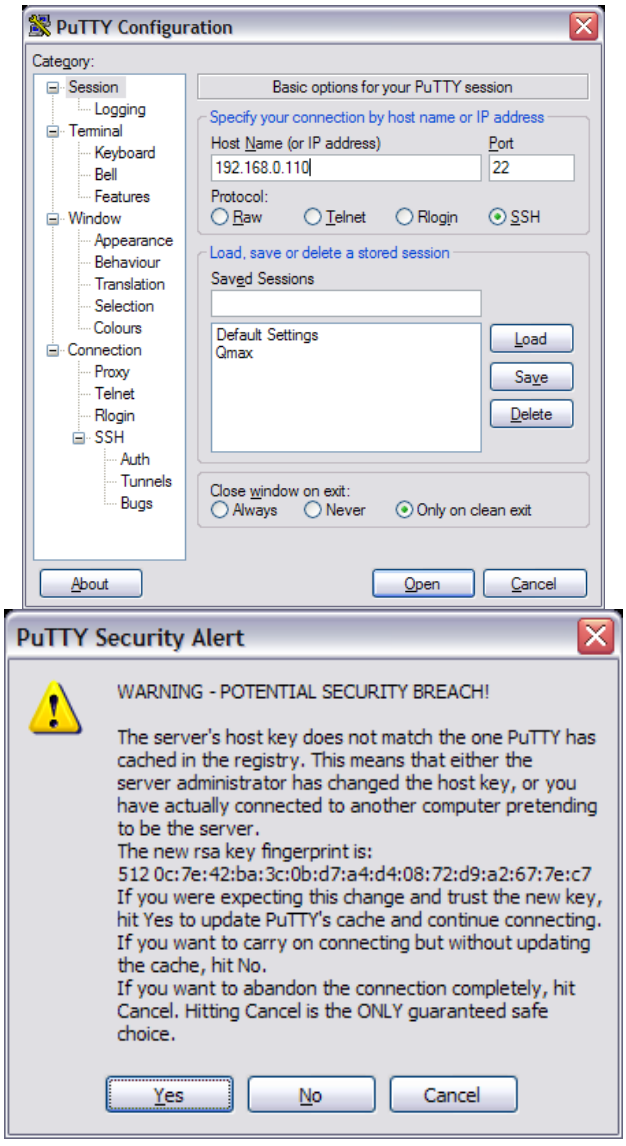


Figure 3: **PuTTY connection for SSH**

9. Next, using the **show vty 0** command, verify that SSH is being used, such as:

```
# show line vty 0
*    1 VTY              -    -      -    -    -    335       0     0/0       -

Line 1, Location: "", Type: "xterm"
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600
```

```
Status: Ready, Active, No Exit Banner, Notify Process
Capabilities: none
Modem state: Ready
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
               00:10:00       00:01:00                      none     not set
                            Idle Session Disconnect Warning
                              never
                            Login-sequence User Response
                             00:00:30
                            Autoselect Initial Wait
                              not set
Modem type is unknown.
Session limit is not set.
Time since activation: 00:02:11
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed input transports are ssh.
Allowed output transports are telnet ssh.
Preferred transport is telnet.
No output characters are padded
No special data dispatching characters
```

10. Run Ethereal, and verify that the username and password cannot be viewed.

Is it possible to view the username and password:

11. If necessary, both TELNET and SSH access can be allowed with:

```
# config t
(config)# line vty 0 15
(config-line)# transport input any
```

Is it possible to TELNET and also SSH from each of the nodes:

12. An open session can be a security risk, especially if it is left unattended, as another user could hi-jack the session. Thus a good security tip is to limit the length of time that a session is allowed to stay inactive. In the following the session time-out is set to one minute:

```
# config t
(config)# line vty 0 15
(config-line)# transport input ssh
(config-line)# session-timeout 1
```

and, after one minute of inactivity the session should be closed, such as:

```
User Access Verification

Username:
% Username:  timeout expired!

Connection to host lost.
```

Create a number of SSH sessions, and verify that after one minute of inactivity that the sessions will time-out. Is this verified:

13. Many firewalls block access to lower ports, such as TELNET and FTP, and thus for TELNET/SSH access the port of the server on the access point must be changed. In the following the port is changed to 2000:

```
(config)# ip ssh port 2000 rotary 0
```

Connect to the SSH service using port 2000 (such as shown in Figure 5). Does it connect:

Achieve the same for TELNET access using the 2001 port. Does it connect using the new port? What configuration is used:

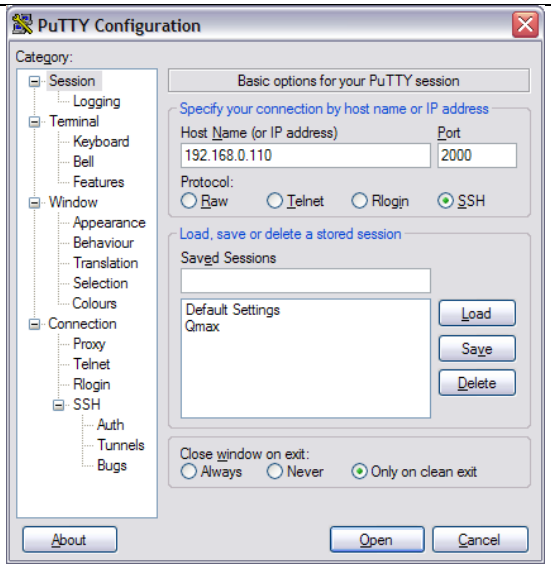


Figure 4: PuTTY connection for SSH

14. Often the administrator wants to limit the number of TELNET sessions. In the following case there is a limit of **three** TELNET/SSH sessions (0, 1 and 2):

```
(config)# line vty 0 2
(config-line)# transport input any
(config)# line vty 3 15
(config-line)# transport input none
```

Connect to the access point with more than three sessions, and verify that it does not allow any more than three. Is it working:

Lab 6: Encryption/Authentication

You will be assigned a group. In this lab the setup is as follows:

Group	Device	SSID	BVI	Host range	Radio channel
A	**Aironet1**	**GroupA**	**10.0.0.1**	**10.0.0.10-10.0.0.12**	**2**
B	**Aironet2**	**GroupB**	**10.0.0.2**	**10.0.0.13-10.0.0.14**	**3**
C	**Aironet3**	**GroupC**	**10.0.0.3**	**10.0.0.15-10.0.0.17**	**4**
D	**Aironet4**	**GroupD**	**10.0.0.4**	**10.0.0.18-10.0.0.19**	**5**
E	**Aironet5**	**GroupE**	**10.0.0.5**	**10.0.0.20-10.0.0.22**	**7**
F	**Aironet6**	**GroupF**	**10.0.0.6**	**10.0.0.23-10.0.0.24**	**8**
G	**Aironet7**	**GroupG**	**10.0.0.7**	**10.0.0.25-10.0.0.27**	**9**

An example setup for GroupA is:

```
hostname GroupA
dot11 ssid GroupA
  authentication open
  guest-mode
int bvi1
  ip address 10.0.0.1 255.255.255.0
interface d0
  channel 2
  station-role root
  ssid GroupA
  no shutdown
interface fa0
  no shutdown
```

2. Setup your wireless network, and ping all the nodes in your network.

Can all the nodes connect to the wireless network, and can ping each other:

Use the command `show dot11 assoc` **on the access point. What is the output:**

2. Setup your access point and nodes (Figure 1) so that they use WEP encryption. An example of the encryption settings for the access point for GroupA could be:

```
hostname ap
int bvi1
  ip address 10.0.0.1 255.255.255.0
exit
dot11 ssid GroupA
  authentication open
  guest-mode
interface d0
 channel 2
 station-role root
```

```
encryption key 1 size 40bit aaaaaaaaaa transmit-key
encryption mode ciphers tkip wep40
ssid APskills
```

Can all the nodes connect to the wireless network, and can ping each other:

Use the command `show dot11 assoc` **on the access point. What is the output:**

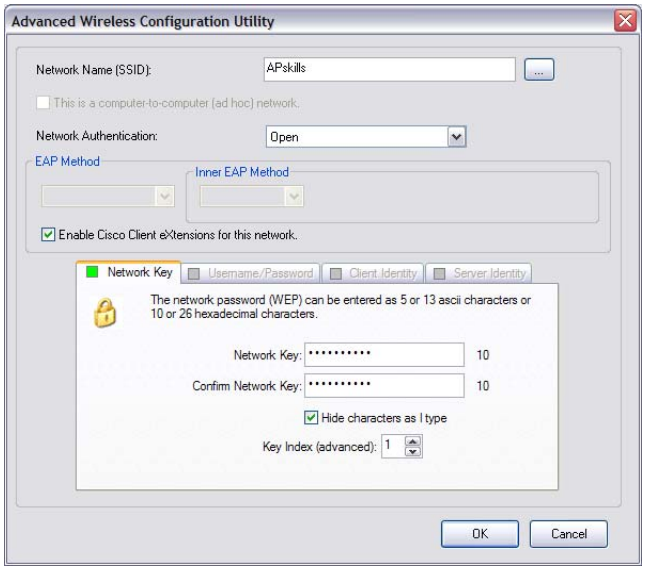


Figure 5: WEP settings

3. Next setup LEAP authentication, with the following (for Group A):

```
hostname ap
aaa new-model
aaa group server radius rad_eap
     server 192.168.1.110 auth-port 1812 acct-port 1813
exit
aaa group server radius rad_mac
exit
aaa group server radius rad_acct
exit
aaa group server radius rad_admin
exit
aaa group server radius dummy
     server 192.168.1.110 auth-port 1812 acct-port 1813
exit
aaa group server radius rad_pmip
exit
aaa authentication login eap_methods group rad_eap
aaa authentication login mac_methods local
aaa authorization exec default local
aaa authorization ipmobile default group rad_pmip
aaa accounting network acct_methods start-stop group rad_acct
aaa session-id common
int bvi1
  ip address 10.0.0.1 255.255.255.0
exit
radius-server local
 nas 10.0.0.1 key sharedkey
```

```
 user aaauser password aaauser
 exit
radius-server host 10.0.0.1 auth 1812 acct 1813 key sharedkey
dot11 ssid GroupA
  authentication open
  authentication network-eap eap_methods
  guest-mode
interface d0
  channel 11
  station-role root
  encryption key 1 size 40bit aaaaaaaaaa transmit-key
  encryption mode ciphers tkip wep40
  ssid GroupA
```

4. Next setup the clients to support LEAP authentication, as shown in Figure 1. Once the client has associated, determine the associated devices with:

```
# show dot assoc

802.11 Client Stations on Dot11Radio0:
SSID [APskills] :

MAC Address    IP address      Device        Name     Parent   State
0090.4b54.d83a 10.0.0.1       4500-radio    -        self     EAP-Assoc

Others:  (not related to any ssid)
```

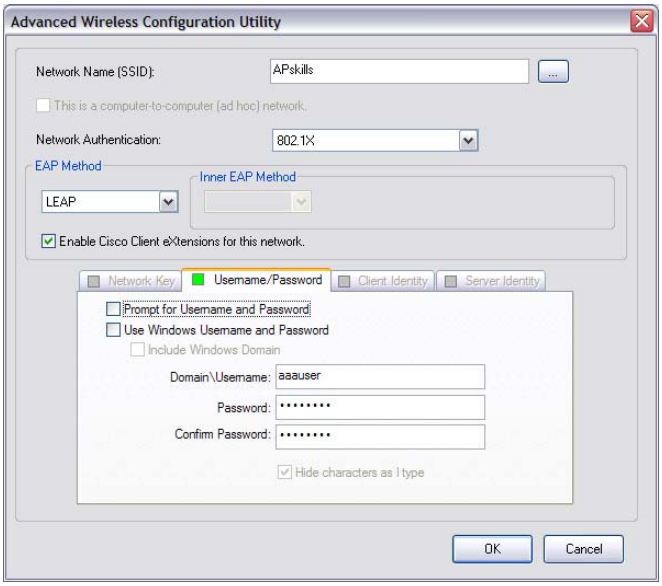


Figure 6: **LEAP setup**

Which devices have associated:

Did you see a message on the access point which had the following format:

*Mar 1 00:00:51.750: %DOT11-6-ASSOC: Interface Dot11Radio0, Station 0090.4b54.d83a Associated KEY_MGMT[WPA]

5. Next setup WPA with TKIP encryption, and LEAP authentication with (for Group A):

```
hostname ap
aaa new-model
aaa group server radius rad_eap
     server 192.168.1.110 auth-port 1812 acct-port 1813
exit
aaa group server radius rad_mac
exit
aaa group server radius rad_acct
exit
aaa group server radius rad_admin
exit
aaa group server radius dummy
     server 10.0.0.1 auth-port 1812 acct-port 1813
exit
aaa group server radius rad_pmip
exit
aaa authentication login eap_methods group rad_eap
aaa authentication login mac_methods local
aaa authorization exec default local
aaa authorization ipmobile default group rad_pmip
aaa accounting network acct_methods start-stop group rad_acct
aaa session-id common
int bvi1
  ip address 10.0.0.1 255.255.255.0
exit
radius-server local
 nas 10.0.0.1 key sharedkey
 user aaauser password aaauser
 exit
radius-server host 10.0.0.1 auth 1812 acct 1813 key sharedkey
dot11 ssid GroupA
  authentication open
  auth key-management wpa
  authentication network-eap eap_methods
  guest-mode
interface d0
  channel 2
  station-role root
  encryption mode ciphers tkip
  ssid GroupA
```

Which devices have associated:

Did you see a message on the access point which had the following format:

*Mar 1 00:00:51.750: %DOT11-6-ASSOC: Interface Dot11Radio0, Station 0090.4b54.d83a Associated KEY_MGMT[WPA]

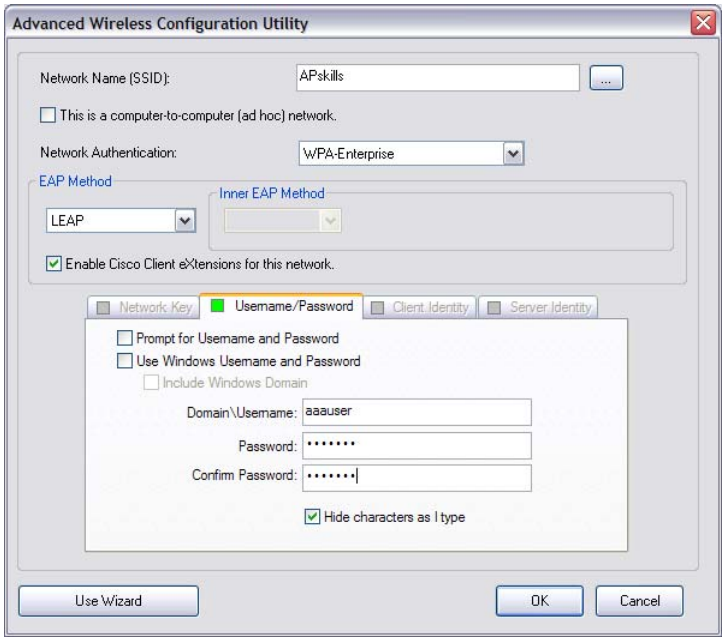


Figure 7: LEAP setup

6. If the client supports CCKM, then the following can be setup (for Group A):

```
hostname ap
aaa new-model
aaa group server radius rad_eap
 server 10.0.0.1 auth-port 1812 acct-port 1813
 exit
aaa authentication login eap_methods group rad_eap
int bvi1
  ip address 10.0.0.1 255.255.255.0
exit
radius-server local
 nas 10.0.0.1 key sharedkey
 user aaauser password aaauser
 exit
radius-server host 10.0.0.1 auth 1812 acct 1813 key sharedkey
dot11 ssid GroupA
  authentication open
  auth key-management cckm
  authentication network-eap eap_methods
  guest-mode
interface d0
 channel 2
 station-role root
 encryption mode ciphers tkip
 ssid GroupA
```

Lab 7: Filtering/Blocking

You will be assigned a group. In this lab the setup is as follows:

Group	Device	SSID	BVI	Host range	Radio channel
A	**Aironet1**	**GroupA**	**192.168.2.1**	**192.168.2.10-192.168.2.12**	**2**
B	**Aironet2**	**GroupB**	**192.168.2.2**	**192.168.2.13-192.168.2.14**	**3**
C	**Aironet3**	**GroupC**	**192.168.2.3**	**192.168.2.15-192.168.2.17**	**4**
D	**Aironet4**	**GroupD**	**192.168.2.4**	**192.168.2.18-192.168.2.19**	**5**
E	**Aironet5**	**GroupE**	**192.168.2.5**	**192.168.2.20-192.168.2.22**	**7**
F	**Aironet6**	**GroupF**	**192.168.2.6**	**192.168.2.23-192.168.2.24**	**8**
G	**Aironet7**	**GroupG**	**192.168.2.7**	**192.168.2.25-192.168.2.27**	**9**

The setup for the Windows server is 192.168.2.8 and the Linux server is 192.168.2.9. A diagram of the system is shown in Figure 1.

The wireless access point can be used to filter mac addresses for a source and destination. Its format is:

```
access-list [deny | permit] [source ac] [source mask] [dest mac] [dest mask]
```

For example to disallow the node with the mac address of 0090.4b54.d83a access to 0060.b39f.cae1:

```
access-list 1101 deny  0090.4b54.d83a 0.0.0 0060.b39f.cae1 0.0.0
access-list 1101 permit 0.0.0  ffff.ffff.ffff 0.0.0  ffff.ffff.ffff
```

and it is applied with the following:

```
int d0
 l2-filter bridge-group-acl
 bridge-group 1
 bridge-group 1 subscriber-loop-control
 bridge-group 1 output-pattern 1101
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding
 bridge-group 1 spanning-disabled
```

```
ap#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.110           -   000d.65a9.cb1b  ARPA   BVI1
Internet  192.168.1.101           1   0060.b39f.cae1  ARPA   BVI1
Internet  192.168.1.103           2   0009.7c85.87f1  ARPA   BVI1
Internet  192.168.1.115           1   0090.4b54.d83a  ARPA   BVI1
ap#
```

Determine all the mac addresses on your network:

Block the access of one computer to another. What is the access-list used:

Is the access blocked, and can the other nodes still access each other:

1. Next remove the access list with:

```
no access-list 1101
```

   and now add a new one which block access from one computer to two of the hosts on the network.

Is the block successful:

2. The access point supports access-lists. For example, the following blocks a host at 192.168.1.111 access to 192.168.1.110:

```
ip access-list extended Test
 deny   ip host 192.168.1.111 host 192.168.1.110
 permit ip any any
dot11 ssid GroupA
  authentication open
  guest-mode
interface d0
 channel 11
 ip access-group Test in
 station-role root
 ssid GroupA
```

3. Create a wireless network which blocks one of the nodes on the network, and allows the other one.

What is the access-list:

Do the blocks work, and can the other nodes still communicate:

4. Along with IP filtering it is possible to filter for the TCP port. For example the following blocking of any source host to any destation on port 80

```
ip access-list extended Test
 deny   tcp any any eq 80
```

```
 permit ip any any
dot11 ssid GroupA
  authentication open
  guest-mode
  end

interface d0
 channel 11
 ip access-group Test in
 station-role root
 ssid GroupA
```

5. Test the above script and make sure that none of the nodes can access the web server on the access point:

Is web access blocked:

6. Modify the access-list so that only one node is blocked access to the web server on the access point:

Is web access blocked:

3. Using the client and the server program, write and access-list which will block communications between two of the nodes on the network for client-server communications on port 1001:

Is the access blocked:

4. It is possible to block ICMP in the filtering, such as blocking a ping from 192.168.1.111 to 192.168.1.110:

```
ip access-list extended Test
 deny icmp 192.168.1.111 0.0.0.0 192.168.1.110 0.0.0.0
 permit ip any any
```

Block a ping from one of the nodes on the network to the access point. Can you ping the access point from it? **Can you ping from other nodes in the network?**

5. Block a ping from one of the nodes on your network to another node.

Is it possible to ping the access-point from one of the nodes:

Is it possible to ping from one of the nodes to the other:

Can you ping the Windows server?

Lab 8: VLAN

The access point can assign VLANs, where the nodes in the same VLAN can connect to each other, but cannot communicate directly with nodes on another VLAN. This allows nodes to connect to each other, even though they connect to a different access device. In a wireless system the nodes can communicate with a VLAN over different SSID. The mechanism used is **IEEE 802.1Q** tagging. The setup for the lab is defined in Figure 1.

Thus, now setup the following:

SSID Group 1: MyVLAN1a, MyVLAN2a
SSID Group 2: MyVLAN1b, MyVLAN2c
SSID Group 3: MyVLAN1c, MyVLAN2c

PC1-PC5: 192.168.0.1-5
Access point: 192.168.0.100

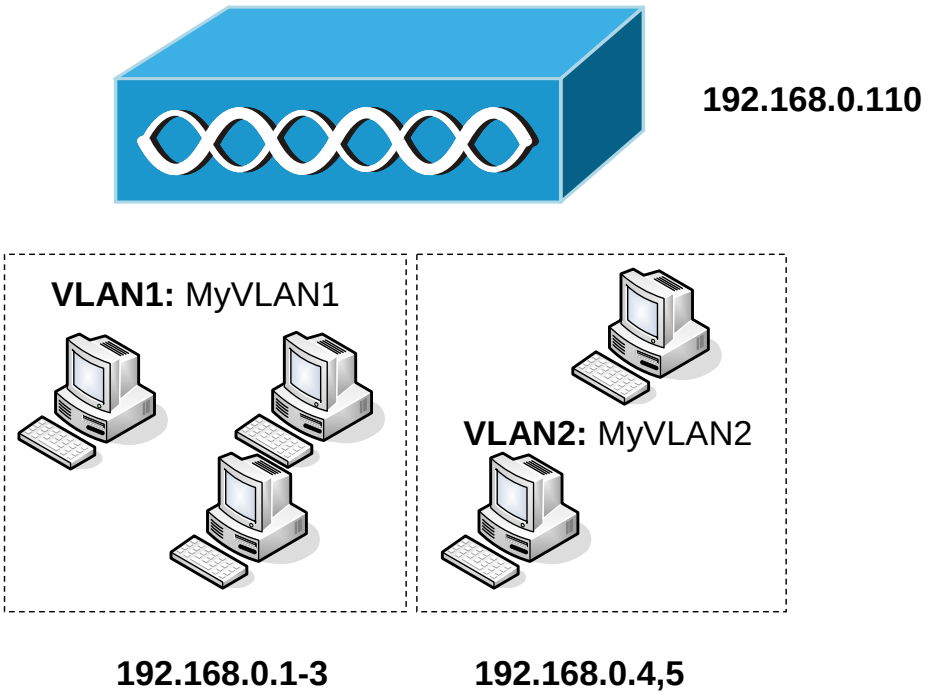


Figure 8: **Wireless configuration**

Nodes PC1, PC2 and PC3 should associate with **MyVLAN1**, and PC4 and PC5 should connect to **MyVLAN2**. Assign the MyVLAN1 SSID to VLAN 1 and MyVLAN2 SSID to VLAN 2.

Can nodes PC1, PC2 and PC3 ping each other:
Can nodes PC4 and PC5 ping each other:

Show that PC4 and PC5 cannot communicate with PC1, PC2, and PC3.

What are the associations:

An example of the configuration for Group 1 is:

```
(config)# interface BVI1
(config-if)# ip address 192.168.0.110 255.255.255.0
(config)# interface Dot11Radio0
(config-if)# encryption key 1 size 40bit aaaaaaaaaa transmit-key
(config-if)# encryption mode ciphers tkip wep40
(config-if)# ssid APskills1
(config-ssid)#  authentication open
(config-ssid)#  guest-mode
(config-ssid)# ssid MyVLAN1a
(config-ssid)# vlan 1
(config-ssid)# authentication open
(config-ssid)# ssid MyVLAN2a
(config-ssid)# authentication open
(config-ssid)# vlan 2
```

6. Now configure the sub-interfaces for the radio port and define IEEE 802.11Q tagging, and assign them to a bridge group:

```
(config)# interface Dot11Radio0.1
(config-if)# encapsulation dot1Q 1 native
(config-if)# bridge-group 1
(config-if)# interface Dot11Radio0.2
(config-if)# encapsulation dot1Q 2
(config-if)# bridge-group 2
```

Can nodes PC1, PC2 and PC3 ping each other:

Can nodes PC4 and PC5 ping each other:

Show that PC4 and PC5 cannot communicate with PC1, PC2, and PC3.

What are the associations:

Using **show vlan**, show that the output is in the form:

```
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interfaces:  Dot11Radio0.1
Virtual-Dot11Radio0.1

 This is configured as native Vlan for the following interface(s) :
Dot11Radio0
Virtual-Dot11Radio0
```

```
   Protocols Configured:   Address:              Received:        Transmitted:
        Bridging        Bridge Group 1               17                  9
        Bridging        Bridge Group 1               17                  9

Virtual LAN ID:  2 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interfaces:  Dot11Radio0.2
Virtual-Dot11Radio0.2

   Protocols Configured:   Address:              Received:        Transmitted:
        Bridging        Bridge Group 2                1                  0
        Bridging        Bridge Group 2                1                  0
```

7. Now we will group the VLANs together, if required, with a bridge group. Thus:

```
(config-if)# interface Dot11Radio0.2
(config-if)# no bridge-group 2
(config-if)# bridge-group 1
```

Can nodes PC1, PC2 and PC3 ping each other:

Can nodes PC4 and PC5 ping each other:

Show that PC4 and PC5 can now communicate with PC1, PC2, and PC3.

What are the associations:

Lab 9: VLANs and 802.1Q

The access point can assign VLANs, where the nodes in the same VLAN can connect to each other, but cannot communicate directly with nodes on another VLAN. This allows nodes to connect to each other, even though they connect to a different access device. In a wireless system the nodes can thus communicate with a VLAN over a different SSID. The mechanism used is **IEEE 802.1Q** tagging.

The setup for the lab is defined in Figure 1, and the details are:

Group	Device	SSID	BVI	Host range	Radio channel
A	Aironet1	Scotland (VLAN 1) England (VLAN 2)	10.0.0.4	10.0.0.10-10.0.0.12 10.0.1.1-10.0.1.2	2
B	Aironet2	Ireland (VLAN 1) Wales (VLAN 2)	10.0.0.2	10.0.0.13-10.0.0.15 10.0.1.3-10.0.1.4	3
C	Aironet3	France (VLAN 1) Germany (VLAN 2)	10.0.0.3	10.0.0.16-10.0.0.18 10.0.1.5-10.0.1.6	4
D	Aironet4	USA (VLAN 1) Japan (VLAN 2)	10.0.0.4	10.0.0.19-10.0.0.21 10.0.1.7-10.0.1.8	5

1. Setup the connections, so that the first three nodes (PC1, PC2 and PC3) should associate with the first SSID (such as Scotland), and PC4 and PC5 should connect to the second SSID (such as England).

An outline of the configuration for Group A is:

```
(config)# dot11 ssid Scotland
(config-ssid)# authentication open
(config-ssid)# vlan 1
(config-ssid)# guest-mode
(config-ssid)# exit
(config)# dot11 ssid England
(config-ssid)# authentication open
(config-ssid)# vlan 2
(config-ssid)#exit
(config)# interface BVI1
(config-if)# ip address 192.168.0.110 255.255.255.0
(config)# interface Dot11Radio0
(config-if)# channel 1
(config-if)# ssid Scotland
(config-if)# ssid England
(config-if)# no shutdown
(config-if)# int fa0
(config-if)# no shutdown
```

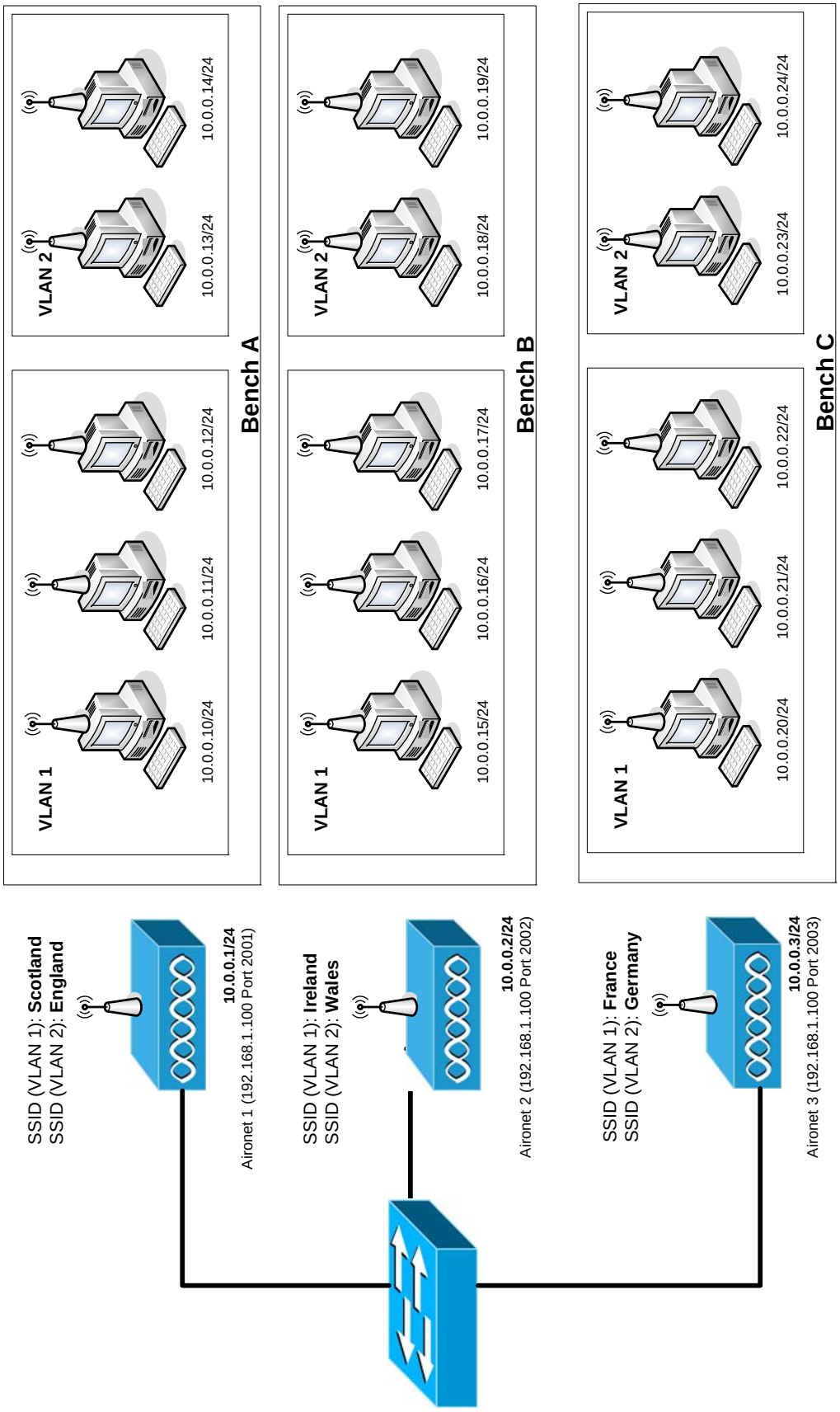


Figure 1: Outline of lab

2. Now configure the sub-interfaces for the radio port and define IEEE 802.1Q tagging, and assign them to a bridge group:

```
(config)# interface Dot11Radio0.1
(config-subif)# ?
Interface configuration commands:
  arp             Set arp type (arpa, probe, snap) or timeout
  bandwidth       Set bandwidth informational parameter
  bridge-group    Transparent bridging interface parameters
  cdp             CDP interface subcommands
  default         Set a command to its defaults
  delay           Specify interface throughput delay
  description     Interface specific description
  encapsulation   Set encapsulation type for an interface
  exit            Exit from interface configuration mode
  ip              Interface Internet Protocol config commands
  keepalive       Enable keepalive
  logging         Configure logging for interface
  mtu             Set the interface Maximum Transmission Unit (MTU)
  no              Negate a command or set its defaults
  service-policy  Configure QoS Service Policy
  shutdown        Shutdown the selected interface
  timeout         Define timeout values for this interface
(config-subif)# encapsulation ?
  dot1Q  IEEE 802.1Q Virtual LAN
(config-subif)# encapsulation dot1q ?
  <1-4094>  IEEE 802.1Q VLAN ID
(config-subif)# encapsulation dot1q 1 ?
  native        Make this as native vlan
  second-dot1q  Configure this subinterface as a 1Q-in-1Q subinterface
  <cr>
(config-if)# encapsulation dot1q 1 native
(config-if)# bridge-group 1
(config-if)# interface Dot11Radio0.2
(config-if)# encapsulation dot1q 2
(config-if)# bridge-group 2
```

Can nodes PC1, PC2 and PC3 ping each other:

Can nodes PC4 and PC5 ping each other:

Show that PC4 and PC5 cannot communicate with PC1, PC2, and PC3.

What are the associations:

Using **show vlan**, show that the output is in the form:

```
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interfaces:  Dot11Radio0.1
Virtual-Dot11Radio0.1
 This is configured as native Vlan for the following interface(s) :
Dot11Radio0
Virtual-Dot11Radio0
   Protocols Configured:   Address:              Received:        Transmitted:
        Bridging        Bridge Group 1               17                   9
        Bridging        Bridge Group 1               17                   9
Virtual LAN ID:  2 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interfaces:  Dot11Radio0.2
Virtual-Dot11Radio0.2
   Protocols Configured:   Address:              Received:        Transmitted:
        Bridging        Bridge Group 2                1                   0
        Bridging        Bridge Group 2                1                   0
```

3. Now we will group the VLANs together, if required, with a bridge group. Thus:

```
(config-if)# interface Dot11Radio0.2
(config-if)# no bridge-group 2
(config-if)# bridge-group 1
```

Can nodes PC1, PC2 and PC3 ping each other:

Can nodes PC4 and PC5 ping each other:

Show that PC4 and PC5 can now communicate with PC1, PC2, and PC3.

What are the associations:

4. The switch which connects the Aironets can be access from:

192.168.1.100 Port 2008

Log into the device, and view the configuration. If 802.1Q trunking is not enhanced, you may need to add the command:

```
(config)# switchport trunk encapsulation dot1q
(config)# interface fa0/1
(config-if)# switchport trunk encapsulation dot1q
```

Now make sure that there is no bridge between the VLANs, and now conduct the following:

Within VLAN 1 which nodes in the whole network can you ping:

Within VLAN 2 which nodes in the whole network can you ping:

Note: In native VLANs, frames in a VLAN are not modified when they are sent over the trunk. Often these are know as *Management VLAN*. These frames will thus be standard Ethernet frames, and have no additional 802.1q information.

Note: To enable multiple SSIDs to be broadcast (add by J.Graves):

```
dot11 ssid TEST1
mbssid guest-mode
dot11 ssid TEST2
mbssid guest-mode
```

then enable mbssid on the radio interface, and then add the SSID's:

```
int Dot11Radio0
mbssid
ssid TEST1
ssid TEST2
```

Lab 10: IP Routing

The access point can assign VLANs, where the nodes in the same VLAN can connect to each other, but cannot communicate directly with nodes on another VLAN. This allows nodes to connect to each other, even though they connect to a different access device. In a wireless system the nodes can thus communicate with a VLAN over a different SSID. The mechanism used is **IEEE 802.1Q** tagging.

The setup for the lab is defined in Figure 1, and the details are:

Group	Device	SSID	BVI	Host range	Radio channel
A	Aironet1	Scotland (VLAN 1) England (VLAN 2)	10.0.0.4	10.0.0.10-10.0.0.12 10.0.1.1-10.0.1.2	2
B	Aironet2	Ireland (VLAN 1) Wales (VLAN 2)	10.0.0.2	10.0.0.13-10.0.0.15 10.0.1.3-10.0.1.4	3
C	Aironet3	France (VLAN 1) Germany (VLAN 2)	10.0.0.3	10.0.0.16-10.0.0.18 10.0.1.5-10.0.1.6	4
D	Aironet4	USA (VLAN 1) Japan (VLAN 2)	10.0.0.5	10.0.0.19-10.0.0.21 10.0.1.7-10.0.1.8	5

CONNECTION WITHIN A VLAN ON A SINGLE ACCESS POINT

1. Setup the connections, so that the first three nodes (PC1, PC2 and PC3) should associate with the first SSID (such as Scotland), and PC4 and PC5 should connect to the second SSID (such as England).

An outline of the configuration for Group A is:

```
# config t
(config)# dot11 ssid Scotland
(config-ssid)# mbssid guest-mode
(config-ssid)# authentication open
(config-ssid)# vlan 1
(config-ssid)# exit

(config)# dot11 ssid England
(config-ssid)# mbssid guest-mode
(config-ssid)# authentication open
(config-ssid)# vlan 2
(config-ssid)# exit

(config)# int BVI1
(config-if)# ip address 10.0.0.4 255.255.255.0
(config-if)# no shut
(config-if)# exit

(config)# int d0
(config-if)# mbssid
(config-if)# ssid Scotland
(config-if)# ssid England
(config-if)# channel 1
(config-if)# no shutdown
(config-if)# exit

(config)# int fa0
(config-if)# no shutdown
(config-if)# exit
```

```
(config)# int d0.1
(config-subif)# ?
Interface configuration commands:
  arp             Set arp type (arpa, probe, snap) or timeout
  bandwidth       Set bandwidth informational parameter
  bridge-group    Transparent bridging interface parameters
  cdp             CDP interface subcommands
  default         Set a command to its defaults
  delay           Specify interface throughput delay
  description     Interface specific description
  encapsulation   Set encapsulation type for an interface
  exit            Exit from interface configuration mode
…
  service-policy  Configure QoS Service Policy
  shutdown        Shutdown the selected interface
  timeout         Define timeout values for this interface
(config-subif)# encapsulation ?
  dot1Q  IEEE 802.1Q Virtual LAN
(config-subif)# encapsulation dot1q ?
  <1-4094>  IEEE 802.1Q VLAN ID
(config-subif)# encapsulation dot1q 1 ?
  native        Make this as native vlan
  second-dot1q  Configure this subinterface as a 1Q-in-1Q subinterface
  <cr>
(config-if)# encapsulation dot1q 1 native
(config-if)# int fa0.1
(config-if)# encapsulation dot1q 1 native
(config-if)# exit
(config-if)# int d0.2
(config-if)# encapsulation dot1q 2
(config-if)# bridge-group 2
(config-if)# int fa0.2
(config-if)# encapsulation dot1q 2
(config-if)# bridge-group 2
(config-if)# exit
```

Can nodes PC1, PC2 and PC3 ping each other:

Can nodes PC4 and PC5 ping each other:

Show that PC4 and PC5 cannot communicate with PC1, PC2, and PC3.

What are the associations:

2. Using **show vlan**, show that the output is in the form:

```
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interfaces:  Dot11Radio0.1
Virtual-Dot11Radio0.1
 This is configured as native Vlan for the following interface(s) :
Dot11Radio0
Virtual-Dot11Radio0
   Protocols Configured:   Address:              Received:        Transmitted:
        Bridging        Bridge Group 1                17                   9
        Bridging        Bridge Group 1                17                   9
Virtual LAN ID:  2 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interfaces:  Dot11Radio0.2
Virtual-Dot11Radio0.2
   Protocols Configured:   Address:              Received:        Transmitted:
        Bridging        Bridge Group 2                 1                   0
        Bridging        Bridge Group 2                 1                   0
```

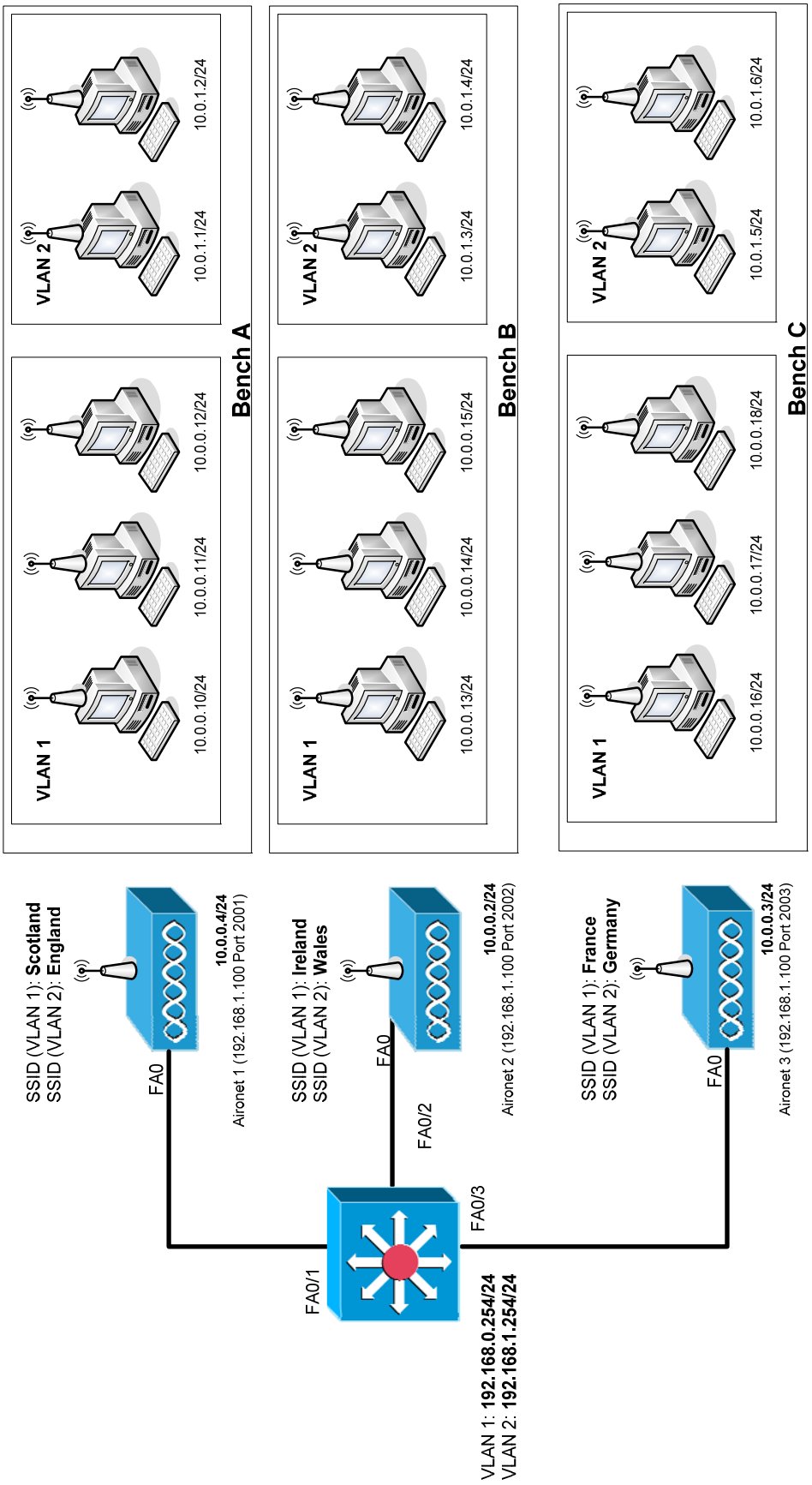


Figure 1: Outline of lab

3. Now we will group the VLANs together, if required, with a bridge group. Thus:

```
(config-if)# interface Dot11Radio0.2
(config-if)# no bridge-group 2
(config-if)# bridge-group 1
```

Can nodes PC1, PC2 and PC3 ping each other? Can nodes PC4 and PC5 ping each other?

Show that PC4 and PC5 can now communicate with PC1, PC2, and PC3. What are the associations:

Objective: You should be able to access the other VLAN on the same access point.

4. Now reassign the bridge-groups, such as:

```
(config-if)# interface Dot11Radio0.2
(config-if)# no bridge-group 1
(config-if)# bridge-group 2
```

Objective: You should not be able to access the other VLAN on the same access point.

ENABLING TRUNKING BETWEEN VLANs

5. The switch which connects the Aironets can be accessed from 192.168.1.100 Port 2008. Log into the device, and view its configuration. 802.1q can be enabled and trunked between the ports of the switch with:

```
# vlan database
(vlan)# vlan 1
(vlan)# vlan 2
(vlan)# exit
# config t
(config)# int fa0/1
(config-if)# switchport trunk encapsulation dot1q
(config-if)# switchport trunk native vlan 1
(config-if)# switchport trunk allowed vlan 1,2
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
(config-if)# int fa0/2
(config-if)# switchport trunk encapsulation dot1q
(config-if)# switchport trunk native vlan 1
(config-if)# switchport trunk allowed vlan 1,2
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
(config-if)# int fa0/3
(config-if)# switchport trunk encapsulation dot1q
(config-if)# switchport trunk native vlan 1
(config-if)# switchport trunk allowed vlan 1,2
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
(config-if)# int fa0/4
(config-if)# switchport trunk encapsulation dot1q
(config-if)# switchport trunk native vlan 1
(config-if)# switchport trunk allowed vlan 1,2
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
(config)# exit
# exit
```

6. Now make sure that there is no bridge between the VLANs, and now conduct the following:

Within VLAN 1 which nodes in the whole network can you ping:

Within VLAN 2 which nodes in the whole network can you ping:

All the nodes in VLAN 1 should be able to ping each other.

All the nodes in VLAN 1 should be able to ping each other.

Nodes in VLAN 1 cannot ping nodes in VLAN 2, and vice-versa.

Objective: You should be able to ping any node in your VLAN, no matter which access point they connect to, but not in other VLANs. PLEASE NOTE IT CAN TAKE UP TO A MINUTE FOR THE TRUNKING TO OCCUR … PLEASE BE PATIENT!

ENABLING IP ROUTING BETWEEN VLANs

7. Now we can enable routing between the VLANs, at Layer 3, with modifications on the switch:

```
# config t
(config)# ip routing
(config)# vlan 1
(config-vlan)# exit
(config)# int vlan 1
(config)# ip address 10.0.0.254 255.255.255.0
(config-vlan)# exit
(config)# vlan 2
(config-vlan)# exit
(config)# int vlan 2
(config-if)# ip address 10.0.1.254 255.255.255.0
(config-if)# exit
```

8. Now make sure that you set the default gateway for nodes in VLAN 1 to **10.0.0.254**, and for VLAN 2 to **10.0.1.254**. This will send all the unknown traffic to the switch.

Within VLAN 1 which nodes in the whole network can you ping:

Within VLAN 2 which nodes in the whole network can you ping:

Objective: You should now be able to get the whole network to communicate.

Example configurations

Access Point 1:

```
config t
dot11 ssid Scotland
mbssid guest-mode
authentication open
vlan 1
exit

dot11 ssid England
mbssid guest-mode
authentication open
vlan 2
exit

int BVI1
ip address 10.0.0.4 255.255.255.0
no shut
exit

int d0
mbssid
ssid Scotland
ssid England
channel 1
no shut
exit


int fa0
no shut
exit

int d0.1
encapsulation dot1q 1 native
int fa0.1
encapsulation dot1q 1 native
exit
int d0.2
encapsulation dot1q 2
bridge-group 2
int fa0.2
encapsulation dot1q 2
bridge-group 2
exit
```

Access Point 2:

```
config t
dot11 ssid Ireland
mbssid guest-mode
authentication open
vlan 1
exit

dot11 ssid Wales
mbssid guest-mode
authentication open
vlan 2
exit

int BVI1
ip address 10.0.0.5 255.255.255.0
no shut
exit
int d0
mbssid
ssid Ireland
ssid Wales
channel 2
no shut
exit

int fa0
no shut
exit

int d0.1
encapsulation dot1q 1 native
int fa0.1
encapsulation dot1q 1 native
exit
int d0.2
encapsulation dot1q 2
bridge-group 2
int fa0.2
encapsulation dot1q 2
bridge-group 2
exit
```

Switch configuration

```
vlan database
vlan 1
vlan 2
```

```
exit
config t
int fa0/1
switchport trunk encapsulation dot1q
switchport trunk native vlan 1
switchport trunk allowed vlan 1,2
switchport mode trunk
switchport nonegotiate
int fa0/2
switchport trunk encapsulation dot1q
switchport trunk native vlan 1
switchport trunk allowed vlan 1,2
switchport mode trunk
switchport nonegotiate
int fa0/3
switchport trunk encapsulation dot1q
switchport trunk native vlan 1
switchport trunk allowed vlan 1,2
switchport mode trunk
switchport nonegotiate
int fa0/4
switchport trunk encapsulation dot1q
switchport trunk native vlan 1
switchport trunk allowed vlan 1,2
switchport mode trunk
switchport nonegotiate
exit
exit
```

IP Routing on switch

```
config t
ip routing
int vlan 1
ip address 10.0.0.254 255.255.255.0
no shutdown
int vlan 2
ip address 10.0.1.254 255.255.255.0
no shutdown
```

Lab 11: RADIUS

This lab will show you how to set up a Remote Authentication Dial In User Services (RADIUS) server. The software used in this lab is called FreeRadius, and is a Windows port of the popular RADIUS server for Linux. It can be downloaded at **http://www.freeradius.net/.**

Although you have demonstrated an AP's capability to authenticate to a RADIUS server, this service was on the access point itself. The following procedure highlights the manner in which a RADIUS server can be located remotely, and still provide authentication.

There are two main components to this – the access point and the radius server. Due to topology complications, the RADIUS server will be set up on the same access point as the authentication point. This will be achieved by using two SSID's – one for the RADIUS server to connect to, and one for the clients to connect and authenticate too. This is not common practice. Normally, a RADIUS server will be located somewhere else in the infrastructure, and on a wired link. Please refer to Figure 1.

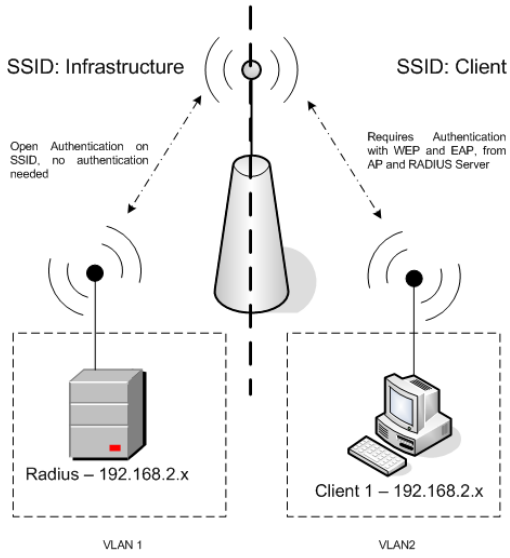


Figure 1

The setup for the lab is defined in Figure 1, and the details are:

Group	Device	SSID	BVI	Host range	Radio channel
A	Aironet1	InfrastructureA (VLAN 1) ClientA (VLAN 2)	192.168.2.1	192.168.2.10- 192.168.2.14	2
B	Aironet2	InfrastructureB (VLAN 1) ClientB (VLAN 2)	192.168.2.2	192.168.2.15 – 192.168.2.19	3
C	Aironet3	InfrastructureC (VLAN 1) ClientC (VLAN 2)	192.168.2.3	192.168.2.20 – 192.168.2.24	4
D	Aironet4	InfrastructureD (VLAN 1) ClientD (VLAN 2)	192.168.2.4	192.168.2.25 – 192.168.2.29	5

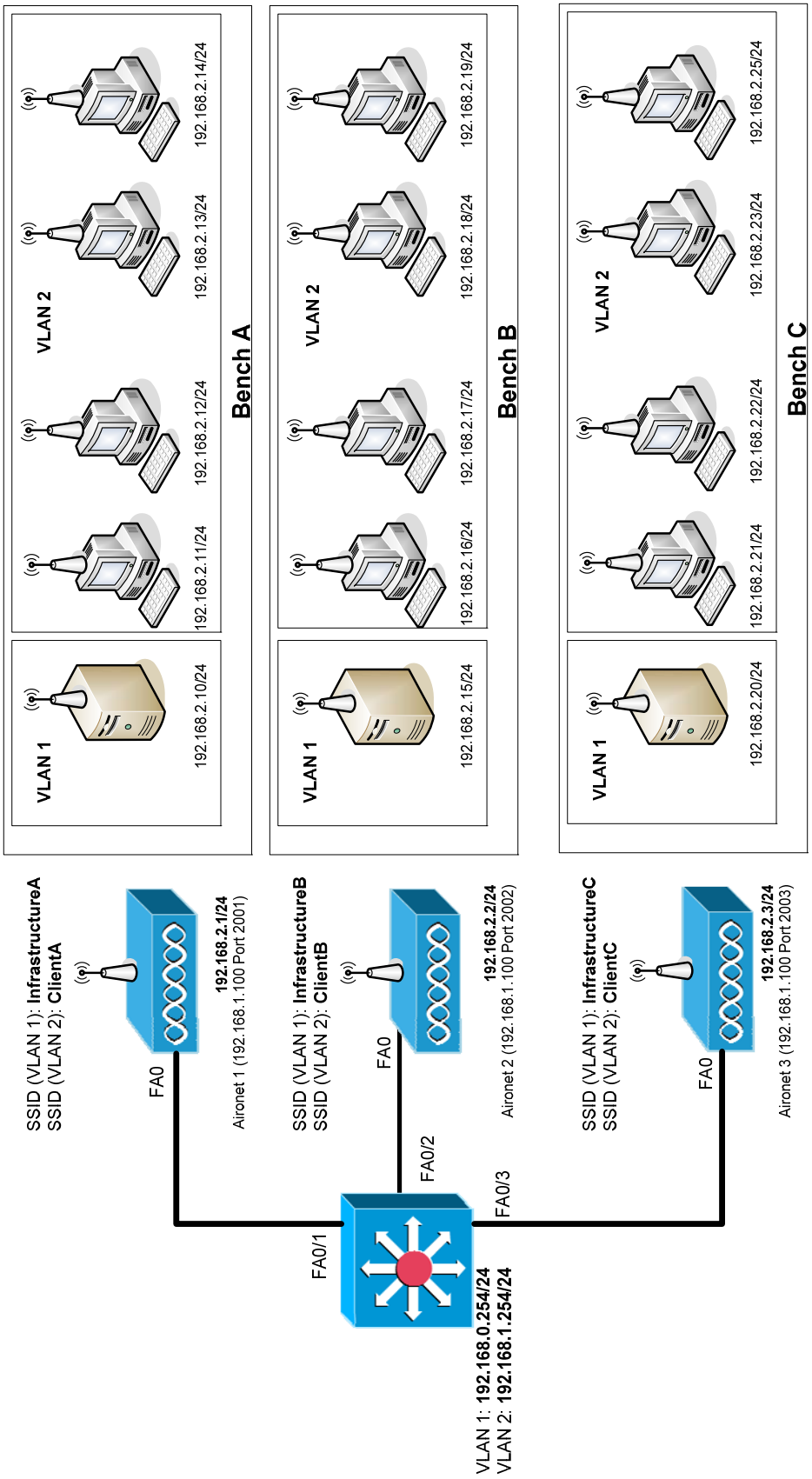


Figure 1: Setup1. Configure the AP with the following commands. Note, erase the startup-config

initially, and re-boot. An outline for Group A is as follows:

```
int d0.1
     encapsulation dot1q 1 native
     bridge-group 1
exit

int d0.2
     encapsulation dot1q 2
     bridge-group 1
exit

dot11 ssid InfrastructureA
     mbssid guest-mode
     authentication open
     vlan 1
exit

dot11 ssid ClientA
     mbssid guest-mode
     authentication network-eap eap_methods
     vlan 2
exit

int BVI1
     ip address 192.168.2.1 255.255.255.0
exit
int d0
     mbssid
     ssid InfrastructureA
     ssid ClientA
     encryption vlan 2 key 2 size 40bit aaaaaaaaaa transmit-key
     encryption vlan 2 mode wep mandatory
     channel 2
     no shut
exit

int fa0
     no shut
exit

aaa new-model
aaa group server radius rad_eap
server 192.168.2.10 auth-port 1812 acct-port 1813
aaa authentication login eap_methods group rad_eap
aaa session-id common
radius-server host 192.168.2.10 auth-port 1812 acct-port 1813 key testing123
```

2. Choose a client to act as the RADIUS server. Connect it to the SSID **Infrastructure**, and assign it the IP address outlined in the previous commands for the RADIUS machine. You can use a machine with a Belkin adaptor for this.

3. Configure the Radius server. You must now configure the Clients.conf file. This is located in **C:\Program Files\FreeRADIUS.net-1.1.1-r0.0.1\etc\raddb** Find a space at the bottom of the document, and add the following:

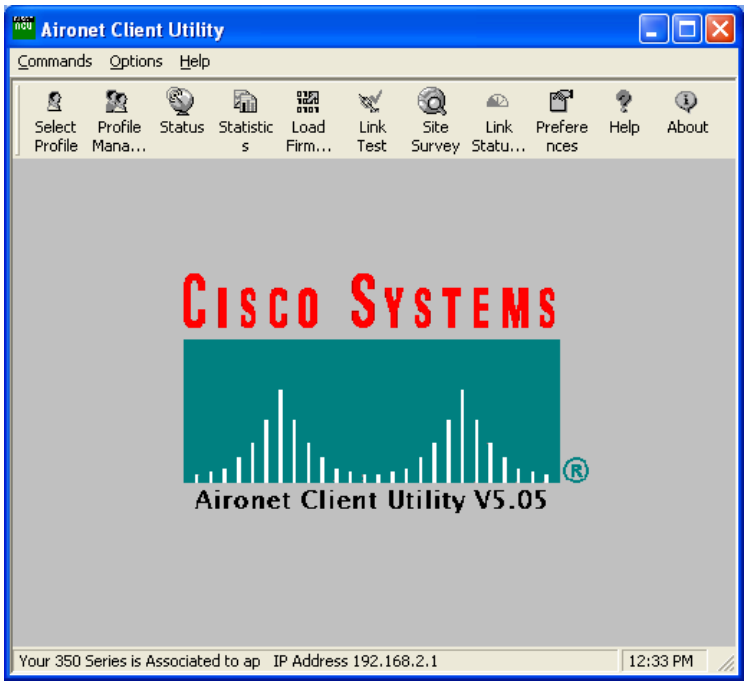
```
client 192.168.2.0/24 {
     secret          = testing123
     shortname       = private-network-2
}
```

4. Start the RADIUS server in debug mode. Debug mode is very useful, as it will inform you of all RADIUS authentication requests, and exactly what it does with

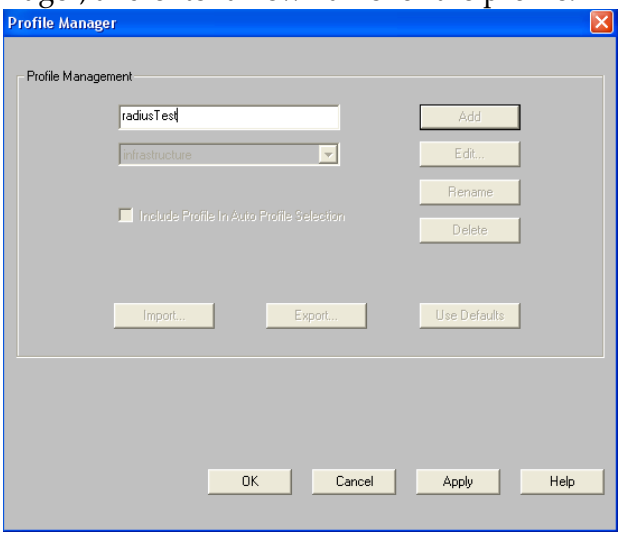
them. You may want to monitor this window when trying to authenticate a machine, to check to see if it works.

```
C:\Program Files\FreeRADIUS.net-1.1.1-r0.0.1\bin> radiusd.exe -d ../etc/raddb –AX
```

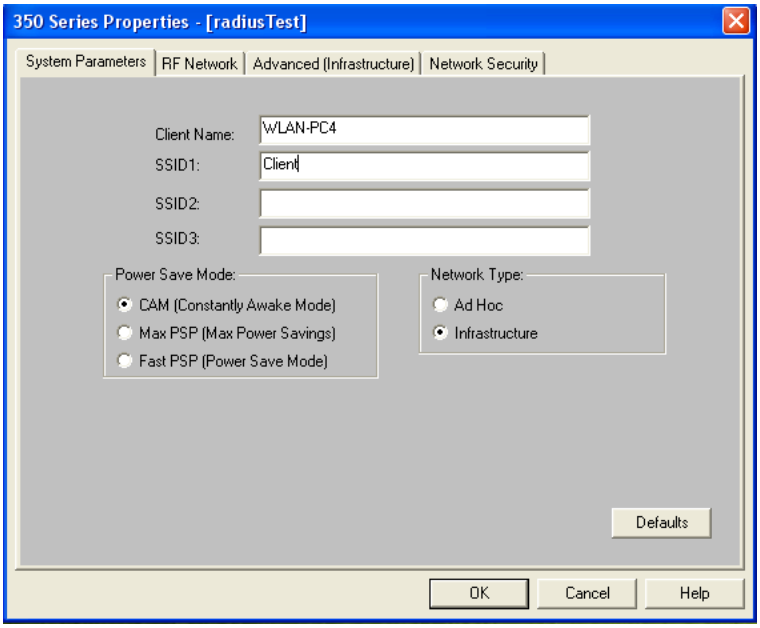
5. Attempt to authenticate the other client to the Client SSID. For this, you will have to use the **Cisco Aironet 350** wireless card in your machine. You must disable the Belkin wireless adaptor for this to work properly. Once you have done so, start the Cisco adaptor, click on the Aironet Client Utility (ACU), and you'll see a screen like this:



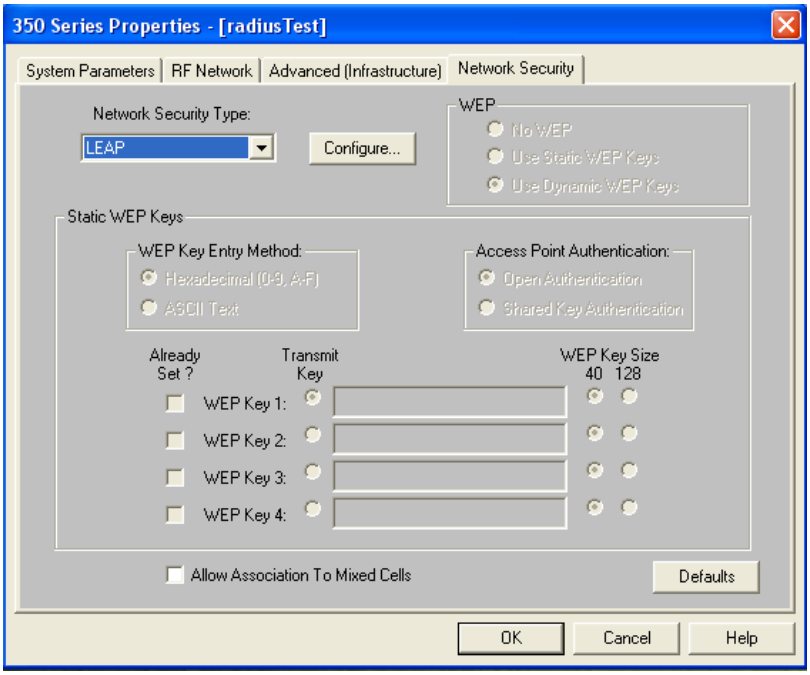
Click on Profile Manager, and enter a new name for the profile:



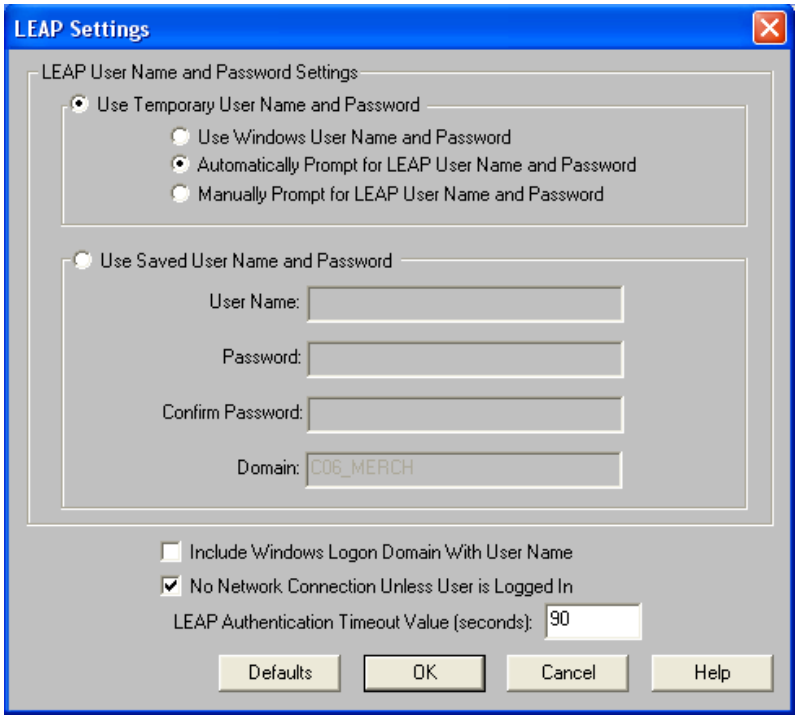
Click OK, and enter your Client ID into the SSID1 field.



Click on the Network Security Tab, and set the screen as follows:



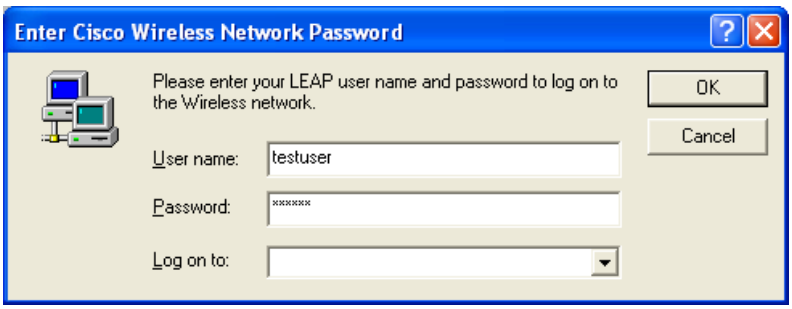
Click on Configure, and set the client as follows:



Click OK, and select your profile. When prompted, enter the user name and password:

```
Username: testuser
Password: testpw
```

Clear the domain box, and click OK. Your main ACU window should display whether you've been successful or not. Once you have a successful authentication, assign an IP address to the adaptor.



6. Show, on the access-point that you have two associations, one should be open and the other should be through EAP-Assoc:

```
ap#show dot11 assoc

802.11 Client Stations on Dot11Radio0:

SSID [ClientA] :

MAC Address    IP address      Device         Name           Parent         State
0009.7cd1.9075 192.168.2.22    350-client     WLAN-PC13      self           EAP-Assoc
```

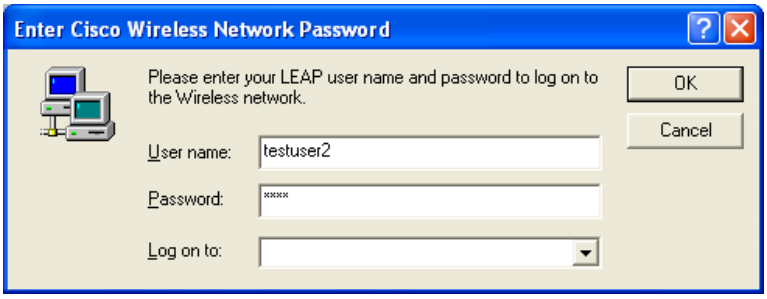
```
SSID [InfrastructureA] :

MAC Address    IP address      Device        Name            Parent         State
0011.5015.b71c 192.168.2.10    4500-radio    -               self           Assoc
```

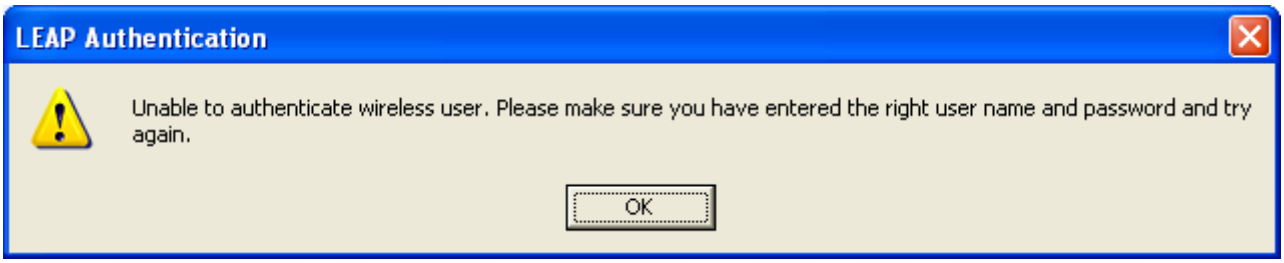
7. Check the details of the RADIUS server, such as:

```
Module: Instantiated radutmp (radutmp)
Listening on authentication *:1812
Listening on accounting *:1813
Ready to process requests.
rad_recv: Access-Request packet from host 192.168.2.1:1645, id=1, length=125
        User-Name = "testuser"
        Framed-MTU = 1400
        Called-Station-Id = "0017.e019.9640"
        Calling-Station-Id = "0009.7cd1.9075"
        Service-Type = Login-User
        Message-Authenticator = 0x5d54508193fc791123ba2fa9e1eccb76
        EAP-Message = 0x0202000d017465737475736572
        NAS-Port-Type = Wireless-802.11
        NAS-Port = 265
        NAS-IP-Address = 192.168.2.1
  Processing the authorize section of radiusd.conf
```

8. Next enter an incorrect user ID, such as:



And show that the authentication is unsuccessful, such as:



9. Explore the **users** file, which is kept in the **C:\Program Files\FreeRADIUS.net-1.1.1-r0.0.1\etc\raddb** directory. Try and add a few new users. Make sure you restart the RADIUS server. An example of a new user is:

```
testuser    User-Password == "testpw"
bill        User-Password == "bill"
```

Can you authenticate using the new users you've added?

10. Use Ethereal to monitor the packets arriving at the RADIUS server. Look at the RADIUS debug screen at the same time, and determine what's happening.

What does it say when you enter an incorrect username and password?

What happens when you change the server secret in the Clients.conf file?

Now monitor Ethereal while supplying correct and false information.

Can you identify the handshake process?

How does it differ when incorrect information is supplied to the RADIUS server?

11. Disable mandatory WEP on VLAN2.

Can you connect to the Radius Server now?

Why Not?

Lab 12: SNMP and Logging

The setup for the lab is:

Group	Device	SSID	BVI	Host range	Radio channel
A	Aironet1	Scotland	10.0.0.4	10.0.0.10-10.0.0.14	2
B	Aironet2	England	10.0.0.2	10.0.0.15-10.0.0.19	3
C	Aironet3	Ireland	10.0.0.3	10.0.0.20-10.0.0.24	4
D	Aironet4	Wales	10.0.0.5	10.0.0.25-10.0.0.29	5

1. Once you have set the network up, install **NetSNMP** on the Windows machines. Enable SNMP on the Aironet with the commands:

```
(config)# snmp-server community public
(config)# snmp-server contact YOURNAME
(config)# snmp-server location C6 lab bench A
(config)# snmp-server chassis-id napier
```

2. Perform an SNMP walk on your Aironet:

```
C:\usr\bin> snmpwalk -Os -c public -v 1 10.0.0.4
sysDescr.0 = STRING: Wireless-G ADSL Gateway
sysObjectID.0 = OID: enterprises.3955.1.1
...
```

SNMP Version

Community string

and determine the following:

System Description:
MAC address of the E0 port:
MAC address of the D0 port:
Up time (s):
Contact name:
MTU (Ethernet):
MTU (D0):
Speed (D0):
IP address (BVI1):

2. Now perform an **smnpget** command to retrieve the values, such as:

```
C:\usr\bin> snmpget -Os -c public -v 1  10.0.0.4 system.sysDescr.0
sysDescr.0 = STRING: Wireless-G ADSL Gateway
```

3. Now use the **snmpwalk** command to view the contents of the tables in the MIB, such as:

```
C:\usr\bin> snmpwalk -Os -c public -v 1 10.0.0.4 system
sysDescr.0 = STRING: Wireless-G ADSL Gateway
sysObjectID.0 = OID: enterprises.3955.1.1
```

```
sysUpTimeInstance = Timeticks: (198354) 0:33:03.54
sysContact.0 = STRING: Linksys
sysName.0 = STRING: Linksys WAG54G
sysLocation.0 = STRING:
sysServices.0 = INTEGER: 4
```

Outline some of the contents of:

SYSTEM:

IF:

ICMP:

TCP:

4. Now change the community string to your Napier:

```
(config)# snmp-server community Napier
(config)# snmp-server contact YOURNAME
(config)# snmp-server location C6 lab
(config)# snmp-server chassis-id napier
```

Which command would you now use to show the SYSTEM table:

5. Ping the Aironet. Now determine the entries which shows the ping:

```
C:\usr\bin> snmpwalk -Os -c public -v 1 10.0.0.4 icmp
icmpInMsgs.0 = Counter32: 14
icmpInErrors.0 = Counter32: 0
icmpInDestUnreachs.0 = Counter32: 2
. . .
```

Which entry defines the count for pings:

6. Enable SNMP on the Windows PC. Now determine:

System Description:
MAC address of the E0 port:
Up time (s):
Contact name:
MTU (Ethernet):
IP address (Ethernet):
Software installed:

Which TCP ports are listening:

Check these with the netstat –a command.

7. Now use the **snmpwalk** command to view the full details of the tables in the MIB, such as:

```
C:\usr\bin> snmpwalk -c public -v 1 10.0.0.10 system
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 6 Model 13 Stepping 8
     AT/AT COMPATIBLE - Software: Windows 2000 Version 5.1 (Build 2600
     Uniprocessor Free)
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.1
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (662239) 1:50:22.39
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: BILL-93D44FD838
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 76
IF-MIB::ifNumber.0 = INTEGER: 5
IF-MIB::ifIndex.1 = INTEGER: 14
```

8. Now use **snmptranslate** to determine the OID numbers:

```
C:\usr\bin> snmptranslate -On SNMPv2-MIB::sysDescr.0
.1.3.6.1.2.1.1.1.0
```

Determine the OID for the following:

System name:
Up time:
Contact name:
Location:
Physical address (Ethernet card):
Physical address (Wireless card):
IP address (Ethernet card):
IP address (Wireless card):
Explain the format of the OID:

9. Enable SNMP on the switch, and determine the following:

System Description:
MAC address of the FA0/1 port:
Up time (s):
Contact name:

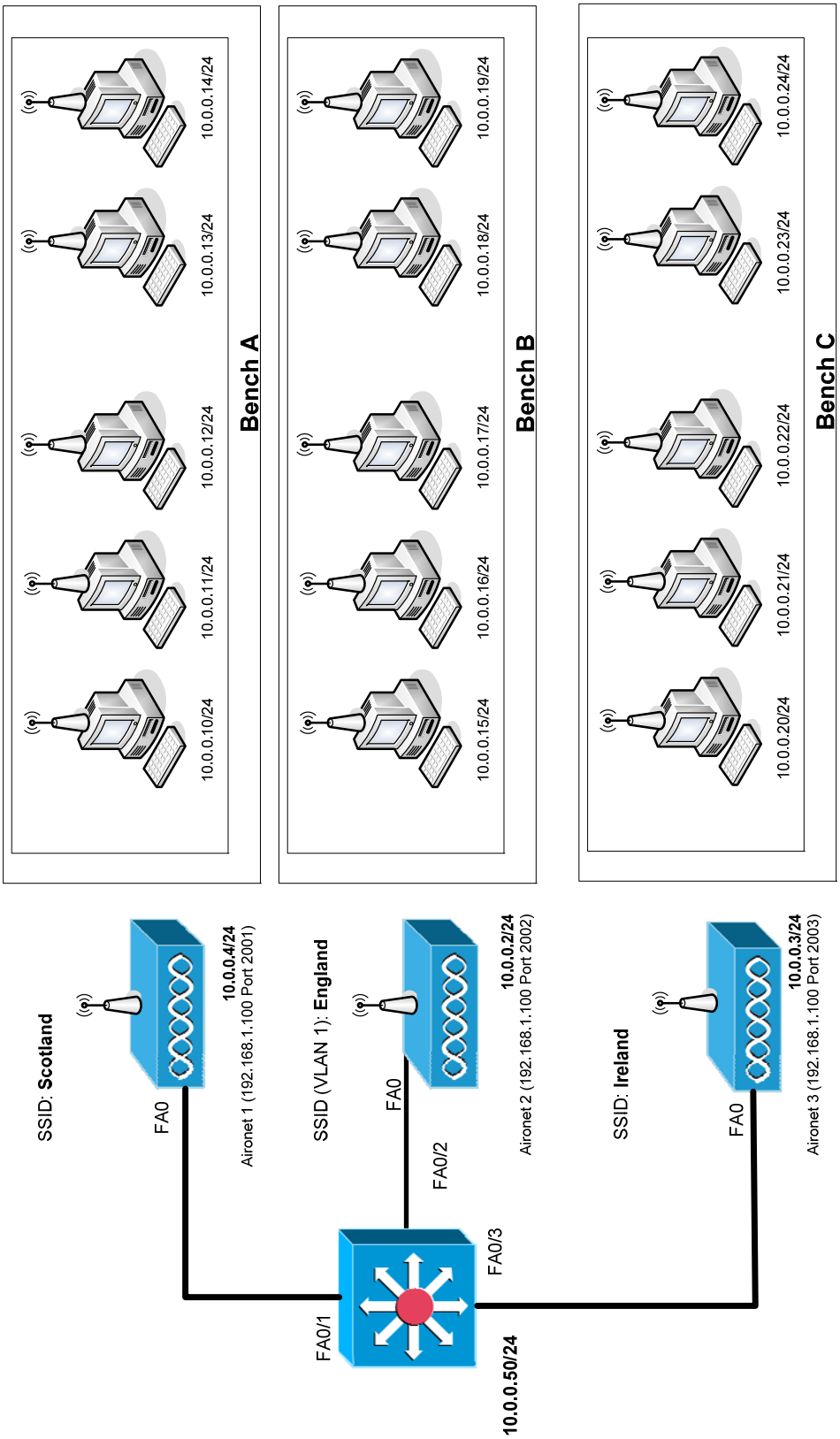


Figure 1: Outline of lab

10. Once you have set the network up, install the **NapierSNMP program** on the Windows machines. Now, using the client, view the SNMP information on the hosts, and also on the Aironet.

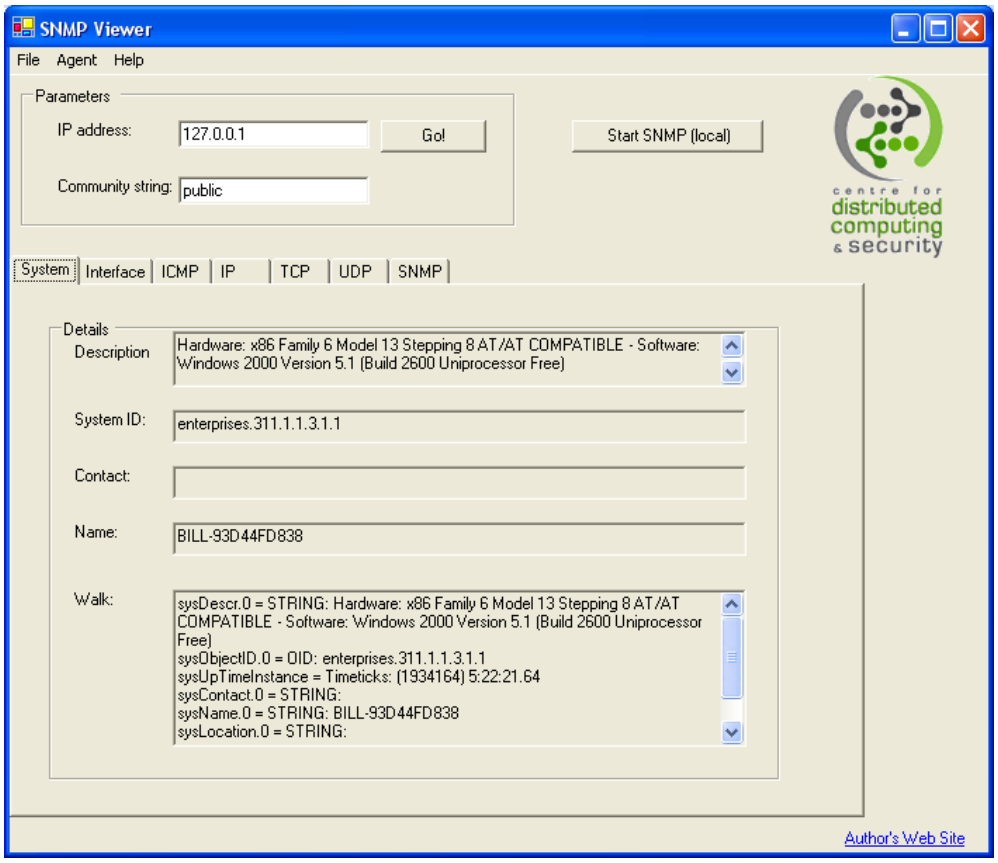


Figure 2:

Note some of the details:

2.2 Logging

The use of logging is important in most networks, especially where there are multiple devices. One method is to use a Syslog server, which can gather the alerts from devices on the network. Along with this, this lab will investigate the TELNET protocol, which is seen as being insecure as the password and user ID of the user is passed through the data packet in plain text. The main objectives are:

1. The usage of logging is important in most networks, especially where there are multiple devices. One method is to use a **Syslog** server, which can gather the alerts from devices on the network. First install the **Kiwi Syslog** program

on all the clients on the network (such as on 10.0.0.10), and start the service with:

```
Manage-> Install the Syslogd service
Manage-> Start the Syslogd service
```

2. Next, enable logging to the Syslog server for each of the nodes with:

```
# config t
(config)# logging 10.0.0.10
(config)# logging 10.0.0.11
(config)# logging 10.0.0.12
(config)# logging 10.0.0.13
(config)# logging 10.0.0.14
```

3. Once it has been setup, verify the operation of the Syslog server by typing in commands, and prompting messages, such as shown in Figure 3.

Do you receive messages on the Syslog server on all the nodes:

Disable logging to 10.0.0.13. Do the messages stop apprearing on this node:

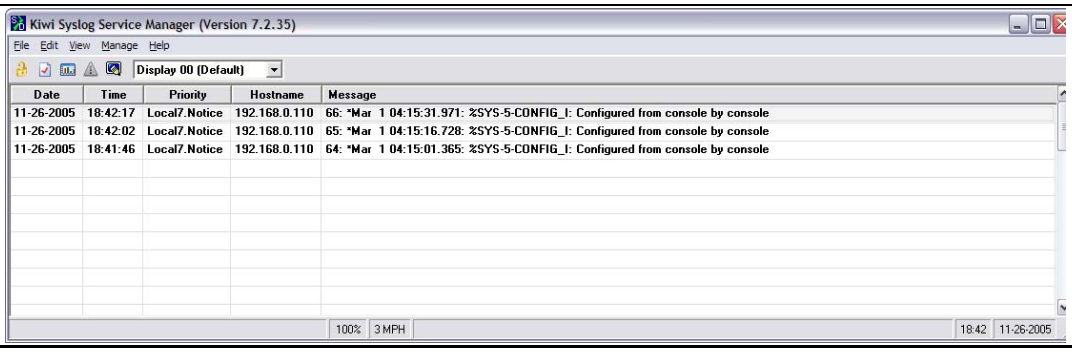


Figure 3: Syslog server

4. The remote login is a source of insecurity, and often the device is setup so that only certain devices can login into the access-point. In the following example, a single device (10.0.0.10) is only allowed access to TELNET into the access point:

```
(config)# access-list 1 permit 10.0.0.10
(config)# access-list 1 deny   any
(config)# line vty 0 15
(config-line)# access-class 1 in
```

Setup the access point so that only one device can login using TELNET. Verify it on each of the clients. Does it work:

Modify it so that it excludes just one address (such as 10.0.0.11) from access, but allows any other address. What is the configuration which achieves this:

5. Often there are problems with intruders when they continually try to login. It is possible to log when the deny part of the access-list is fired, such as:

```
(config)# access-list 1 permit 10.0.0.10
(config)# access-list 1 deny   any log
(config)# line vty 0 15
(config-line)# access-class 1 in
```

6. Now, try to login using a device which is barred from TELNET access, and verify with **sh log** that you get a message such as:

```
*Mar   1 00:50:44.077: %SEC-6-IPACCESSLOGS: list 1 denied 192.168.0.1 1
packet
```

Do you get this message:

Setup the access point to send this message to the Syslog server. Is it received correctly:

Modify the access-list so that the Syslog server also receives a message on a successful access. What is the configuration used:

7. Banners are a way to pass a message to users as they login. Typically they are used to display a message-of-the-day, or to inform users of a change of status. In the first example, setup the EXEC banner with:

```
ap(config)#banner exec #
Enter TEXT message.  End with the character '#'.
You have now entered EXEC mode.
Please be careful when you access the device.
Thank you.
#
```

where the # symbol represents the start and end delimiter.

8. Next exit and verify that you get the following message when you login:

```
ap con0 is now available

Press RETURN to get started.
You have now entered EXEC mode.
Please be careful when you access the device.
Thank you.
ap>
```

9. After this change the login banner with:

```
ap(config)#banner login #
Enter TEXT message.  End with the character '#'.
You are accessing the aironet device.
Please try not to change the EXEC password.
Thank you#
```

10. Using a TELNET or SSH session, now login to the device, and determine where the messages are shown.

Which messages do you receive:

11. Setup a network so that users logging into the network receive the following 'message-of-the-day' message:

```
This is a private network maintained by Napier University.
You should only use this network if you are authorized by C&IT.

Use by authorized persons is not allowed.
```

Additional tutorials

12. Setup the previous network. Now change it so that **Warning** messages, and above, are logged. Verify this.
13. Setup a network so that 10.0.0.10 and 10.0.0.11 can access the wireless access point with TELNET, whereas the other nodes cannot. A successful and an unsuccessful login should be logged on the Syslog server.
14. Setup a network so that 10.0.0.10 and 10.0.0.11 cannot access the wireless access point with TELNET, whereas the other nodes cannot. A successful and an unsuccessful login should be logged on the Syslog server.
15. Setup a network so that only **one** SSH session is possible on the wireless access point.
16. Setup a network so that the Syslog server logs all the successful and unsuccessful radio associations.

17. Create a network which allows up to two TELNET sessions with a timeout for each session at one minute, and up to five SSH sessions with a session timeout of two minutes.